Gigamon®

GigaVUE-OS CLI Reference Guide

Product Version: 6.14

Document Version: 1.0

(See Change Notes for document updates.)

Copyright © 2026 Gigamon Inc. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. No part of this publication may be reproduced, transcribed, translated into any language, stored in a retrieval system, or transmitted in any form or any means without the written permission of Gigamon Inc.

Trademark Attributions

Gigamon and the Gigamon logo are trademarks of Gigamon in the United States and/or other countries. Gigamon trademarks can be found at www.gigamon.com/legal-trademarks. All other trademarks are the trademarks of their respective owners.

Gigamon Inc.
3300 Olcott Street
Santa Clara, CA 95054
408.831.4000

Change Notes

When a document is updated, the document version number on the cover page will indicate a new version and will provide a link to this Change Notes table, which will describe the updates.

Product Version	Document Version	Date Updated	Change Notes
6.14	1.0	07/10/2026	The original release of this document with 6.14.00 GA.

Contents

GigaVUE-OS CLI Reference Guide **1**
Change Notes 3
Contents 4
Introducing the GigaVUE-OS Nodes **12**
About the GigaVUE HC Series and GigaVUE TA Series 12
GigaVUE-OS® HC Series Features and Benefits 15
The Gigamon Deep Observability Pipeline 16
GigaVUE-OS® HC Series Features and Benefits 17
Module Numbering 19
White Box Port and Faceplate Labeling 20
Introducing the GigaVUE-OS CLI **22**
Accessing the Command-Line Interface 22
Command-Line Basics 22
Command Line Modes 22
Entering Commands in the CLI 25
Command-Line Syntax – Entering Commands 26
The Basic Commands 28
What Is Saved In a Configuration File 29
Saving a Configuration File 29
Viewing Saved Configuration Files 30
Using the configuration Command 31
Configuration File Types 31
Syntax for the configuration Command 32
Viewing the Contents of a Configuration File 32
Applying Configuration Files 32
Sharing Configuration Files with Other GigaVUE-OS® HC Series Nodes 33
Command Line Reference **34**
General Information on Working with the CLI 37
Port Lists Definition in the GigaVUE-OS 37
Mode and User Level Commands 38
aaa 38
aaa accounting 39
aaa authentication 40
aaa authorization 44
apps 46

apps asf 47
apps gtp-backup 52
apps exporter 53
apps metadata exporter 56
apps enhanced-slicing 57
apps enhanced asf 58
apps enhanced-lb 60
apps gtp-whitelist 61
apps hsm 65
apps hsm-group 66
apps icap 69
apps inline-ssl 70
apps keystore 86
apps listener 91
apps netflow 92
apps regrex-profile 113
apps sip-whitelist 113
apps split-dns 118
apps ssl 120
apps ssl-profile 123
apps tcp-profile 124
apps trust store 125
banner 126
battery 128
battery optimization 128
battery optimization global 129
bond 130
boot 132
card (GigaVUE-OS® HC Series) 133
card (GigaVUE-OS®TA Series) 136
chassis 137
clear 141
cli 145
clock 147
cluster 148
configuration 154
Configuration File Types 155
configure 161
coreboot 161
crypto 162
debug 169
disable 170
email 170

enable 174
exit 175
fabric advanced-hash 175
file 178
filter-template 180
gigasmart 184
gigastream 188
gigastream advanced-hash 191
gigastream variance-threshold 194
gpfcp profile 195
gsgroup 197
gsparams 200
halt 217
    gsop 217
hb-profile 234
header-strip 237
help 239
hrot 239
hostname 240
ib-pathway 241
image 241
inline-network 243
inline-network-group 249
inline-serial 251
inline-tool 254
inline-tool-group 261
interface 266
ip 269
ip interface 273
ipv6 275
job 279
ldap 284
license 288
map 290
    logging 314
map rule 318
map gsrule 333
map-group 343
map-passall 344
map-scollector 348
map-template 351
nhb-profile 353
no 355

no service ... 356
no traffic ... 357
notifications ... 357
ntp ... 359
ntpdate ... 361
onie ... 361
pcap ... 362
ping ... 367
ping6 ... 367
pld ... 367
proxy-server profile ... 369
policy ... 371
port ... 377
port-group ... 401
port-pair ... 404
ptp ... 405
Use ptp on GigaVUE-TA200 Devices ... 405
radius-server ... 408
redundancy-profile ... 411
reload (reboot) ... 412
reset ... 413
serial ... 414
sfp ... 415
sffp profile ... 416
show ... 417
GigaVUE V Series show ... 433
sleep ... 433
snmp-server ... 434
spine-link ... 454
ssh ... 456
stack-link ... 463
system ... 465
system-health ... 471
tacacs-server ... 472
terminal ... 475
threshold ... 476
tool-mirror ... 478
traceroute ... 479
tunnel ... 480
L2-Circuit Tunnel ... 480
Layer 2 Generic Routing Encapsulation (L2GRE) Tunnel ... 482
Virtual Extensible LAN (VXLAN) Tunnel ... 485
tunnel-endpoint ... 490

uboot ... 492
username ... 492
Access for Read-Only Users ... 494
Change Passwords ... 496
Password Policies ... 496
vport ... 502
web ... 504
write ... 509
GigaVUE-OS CLI—Configuration Examples ... 510
Configure Flow Mapping® ... 510
How to Create Maps ... 511
Configure Shared Collector Maps ... 512
Map Priority ... 514
Adjust Map Priority ... 516
Packets Matching Multiple Rules in Same Map Example ... 517
Port Lists ... 518
How to Add Comments to Map Rules ... 519
Mixing Pass and Drop Rules ... 521
Port Aliases ... 521
User-Defined Pattern Match Rules ... 522
User-Defined Pattern Match Syntax ... 523
User-Defined Pattern Match Rules ... 523
User-Defined Pattern Match Examples ... 525
Map Examples ... 526
How to Handle Overlaps when Sending VLANs and Subnets to Different Tools ... 527
How to Create Map Rules for RTP Traffic ... 528
How to Use MAC Address/Mask Map Rules ... 529
IPv4 Criteria with GigaSMART Operation ... 532
MAC Address Criteria with GigaStream ... 532
IPv6 Criteria ... 533
UDA Pattern Match Criteria ... 533
Null Port in Maps ... 533
How to Create Map Rules with Inner Header Qualifiers and MPLS Header Qualifiers ... 534
map-passalls and port mirrors ... 535
Example of Hybrid Ports ... 539
Port-Filter Examples ... 541
Configure Active Visibility ... 541
Conditions ... 542
Actions ... 546
Policies ... 550
Configure Custom Interface Selection ... 558

Configure GigaStream 559
Regular GigaStream Configuration 560
Controlled GigaStream Configuration 560
Advanced Hashing 561
Weighted GigaStream 561
Configure VLAN Manipulation 561
Configure Ingress and Egress VLAN 564
Ingress Port VLAN Tagging 564
TPID for Ingress Port VLAN Tagging 565
VLAN Tags in Maps 566
Configure Egress Port VLAN Stripping 567
Configure Inline Bypass Solutions 568
Configuration Steps 568
Configure Inline Bypass Examples 572
Configure Inline Bypass Solution on GigaVUE-OS TAP Modules 608
Rules for Inline Bypass on TAP-HC0-G100C0 and TAP-HC1-G10040 608
Example to Configure Inline Bypass on GigaVUE®HC Series Nodes 609
Configure Flexible Inline Arrangements 610
Example 1—Unprotected Flexible Inline, One Collector Map 611
Example 1A—Unprotected Flexible Inline Netlag, One Collector Map 612
Example 2—Unprotected Flexible Inline, Two Collector Maps 614
Example 3—Protected Flexible Inline, Two Collector Maps 616
Example 4—Unprotected Flexible Inline, Rule-Based Map 618
Example 5—Unprotected Flexible Inline, Inline Tool Group 621
Example 6—Unprotected Flexible Inline, Monitoring Mode 623
Example 7—Protected Flexible Inline, Out-of-Band Copy 626
Example 8—Flexible Inline Single Tag Configuration 628
Configure Inline SSL Decryption 628
CLI Configuration Outbound Example 628
CLI Configuration Inbound Example 633
Configure an Inline SSL Session Logging Server Using CLI 634
Configure GigaSMART Operations 635
GigaSMART Operations – Example 635
Configure GigaSMART Masking 637
Configure Packet Slicing 638
GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel) 638
GigaSMART IP Encapsulation (GigaSMART Tunnel) 642
GigaSMART Layer 2 GRE Tunnel Encapsulation/Decapsulation 642
IP Fragmentation and Reassembly on L2GRE and GMIP Tunnels 650
Tunnel Health Checks 651
GigaSMART ERSPAN Tunnel Decapsulation 651
GigaSMART VXLAN Tunnel Decapsulation 653
GigaSMART Custom Tunnel Decapsulation 654

GigaSMART Header Addition 656
GigaSMART De-duplication 656
GigaSMART Header Stripping 657
GigaSMART GTP Correlation 663
GigaSMART GTP Whitelisting and GTP Flow Sampling Examples 674
GigaSMART GTP Overlap Flow Sampling Maps 699
GigaSMART GTP Scaling 704
GigaSMART GTP Stateful Session Recovery 712
GigaSMART SIP/RTP Correlation 713
Configure CPN-UPN Communication solution using CLI 718
Configuration of 5G Correlation 726
GigaSMART FlowVUE 731
GigaSMART Adaptive Packet Filtering (APF) 733
GigaSMART Application Session Filtering (ASF) and Buffer ASF 756
GigaSMART NetFlow Generation 770
GigaSMART Load Balancing 789
GigaSMART MPLS Traffic Performance Enhancement 800
GigaSMART Internet Content Adaptation Protocol (ICAP) 801
GigaSMART SSL Decryption for Out-of-Band Tools 803
Thales Luna HSM for SSL Decryption for iSSL 807
Display GigaSMART Statistics 811
GigaSMART Trailers 813
Configure Clustering 815
Clustering a Node Using Layer 3 Out-of-Band Manual Discovery 815
How to Create a Cluster 817
Inband Cluster Management 829
Inband Cluster Management Configuration Examples 831
Configuration Issues to Consider 831
How to Setup Inband Cluster Management on a New Cluster 832
Enable Cluster Management for GigaVUE TA Series Nodes 849
Inband Cluster Management with GigaVUE TA Series (Including a White Box) 850
Set up Inband Cluster Management with GigaVUE-OS TA-100 or GigaVUE-HC3 859
How to Switch from Inband Cluster Management to Out-of-Band 864
How to Switch from Out-of-Band to Inband Cluster Management 865
How to Remove a Node from an IPv4 Cluster and Add it to an IPv6 Cluster and Vice-versa 867
Troubleshooting 868
Handling System Failure in a Cluster Environment 869
Cluster Commands 869
Cluster-Wide and Local Commands 870
Configure External Authentication in a Cluster 871

Cluster Diagnostics ... 872
Configure Multi-Path Leaf and Spine ... 873
CLI Configuration Example ... 874
Configure GigaVUE HC Series Security Options ... 876
IP Filter Chains for Security ... 876
Disable a Serial Console Port ... 881
Configure Role-Based Access: A Summary ... 882
Role-Based Access: Required Permissions by Command ... 883
Role-Based Access: Rules and Notes ... 885
CLI Commands for Role-Based Access ... 885
Admin-Only CLI Commands ... 887
Configure AAA ... 888
IPv6 Configuration Example ... 897
Encrypt Syslog Audit Data ... 899
CLI Parameter Limits ... 903
System Parameters ... 903
User Parameters ... 904
CLI limits Second Level Map Parameters ... 904
CLI limits Maximum Nodes per Cluster ... 904
CLI limits GigaStream Maximums ... 905
CLI limits Map Rule Maximums ... 905
Alias Limitations ... 905
Additional Sources of Information ... 907
Documentation ... 907
How to Download Software and Release Notes from My Gigamon ... 910
Documentation Feedback ... 910
Contact Technical Support ... 911
Contact Sales ... 912
Premium Support ... 912
The VÜE Community ... 912
Glossary ... 913

Introducing the GigaVUE-OS Nodes

This chapter introduces the GigaVUE HC Series Deep Observability Pipeline nodes, describes their features and functions, and provides an orientation to the physical layout of the models. Refer to the following sections for details:

- About the GigaVUE HC Series and GigaVUE TA Series
- GigaVUE-OS® HC Series Features and Benefits

About the GigaVUE HC Series and GigaVUE TA Series

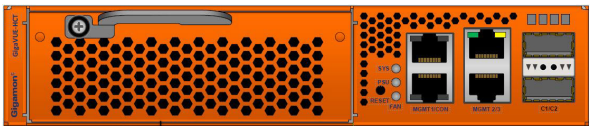
The GigaVUE-OS HC Series delivers performance and intelligence in each of its Deep Observability Pipeline nodes, with port density and speeds that scale to your needs, from 1Gb to 100Gb. With an intuitive Web-based interface (GigaVUE-FM) and a powerful GigaVUE-OS, the Deep Observability Pipeline is able to replicate, filter, and selectively forward network traffic to monitoring, management, and security tools.

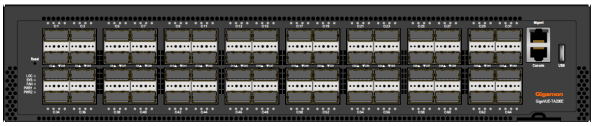
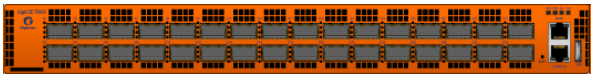
The GigaVUE-OS HC Series and GigaVUE-OS TA Series include the following models that run GigaVUE-OS:

- GigaVUE-HC1
- GigaVUE-HC3
- GigaVUE-HC1-Plus
- GigaVUE-HCT
- GigaVUE-TA25
- GigaVUE-TA25E
- GigaVUE-TA100
- GigaVUE-TA200
- GigaVUE-TA200E
- GigaVUE-TA400
- GigaVUE-TA400E

NOTE: This document describes how to configure and operate the GigaVUE-OS for GigaVUE-OS HC Series and GigaVUE-OS TA Series nodes.

GigaVUE-HC1	• 1RU Footprint • Two Module Slots (Bays) • Dedicated Cluster Management Port • Supports all GigaVUE-HC1 Modules • Cluster with GigaVUE HC Series and GigaVUE-OS TA Series Nodes • All ports, excluding BPS ports, of same type and speed can be used to create GigaStream.	
GigaVUE-HC3	• 3RU Footprint • Four Module Slots (Bays) • Internal Control Card • Extension Board • Dedicated Cluster Management Port • Supports all GigaVUE-HC3 Modules • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes. • All ports, excluding BPS ports, of same type and speed can be used to create GigaStream.	
GigaVUE-HC1-Plus	• 1RU Footprint • Two Module Slots (Bays) and one Fixed Base Module • 4 x 100Gb / 40Gb, 8 x 25Gb, 10Gb, 1Gb connectivity • Supports GigaSMART applications with a fixed rear GigaSMART Module. • Supports Flex Inline in unprotected ports with 10Gb/40Gb/25Gb/100Gb/4 x 10Gb/4x25Gb speed Nodes. • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes • All ports, excluding BPS ports, of same type and speed can be used to create GigaStream.	

GigaVUE-HCT	• 1RU and half rack width system • One Module Slot (Bay) and one Fixed Base Module • 2 x 100Gb/40G bconnectivity • Supports Flex Inline in unprotected ports with 40Gb/100Gb/4x10G/4x25G speed Nodes. • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes • All ports, excluding BPS ports, of the same type and speed can be used to create GigaStream.	
GigaVUE-TA25	• 1 RU Footprint • 1Gb/10Gb/48 x 25Gb/ ports and 8 x 100Gb/40Gb ports, dual hot-pluggable power supplies (AC/DC), four rear hot swappable fan modules, two console ports, and a 10Mb/100Mb/1Gb management port • Optional patch or breakout panel support • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes	
GigaVUE-TA25E	• 1 RU Footprint • 48 x 25Gb/10Gb/1Gb ports and 8 x 100Gb/40Gb port, dual hot-pluggable power supplies (AC/DC), four rear hot swappable fan modules, two console ports, and a 10Mb/100Mb/1Gb management port • Optional patch or breakout panel support • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes.	
GigaVUE-TA100	• 1RU Footprint • 32 x 100Gb/40Gb ports, hot swappable fan modules • Optional patch or breakout panel support • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes.	

GigaVUE-TA200	• 2RU Footprint • 64 x 100Gb/40Gb ports, hot swappable fan modules • Optional patch or breakout panel support • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes.	
GigaVUE-TA200E	• 2RU Footprint • 64 x QSFP28 ports (40G/100G), dual power supply modules (AC/DC) and hot swappable fan modules • UART RS232 Console port (RJ45) and Management port (RJ45).	
GigaVUE-TA400	• 1RU Footprint • 32 x 400Gb /100Gb/ 40Gb QSFP-DD/ QSFP28/QSFP+ ports, dual hot-pluggable power supplies (AC/DC), seven rear hot swappable fan modules, console port and a 10M/100M/1G management port. • Cluster with GigaVUE-OS HC Series and GigaVUE-OS TA Series Nodes.	
GigaVUE-TA400E	▪ 2RU Footprint ▪ 32 40Gb/100Gb/400Gb QSFP-DD/ QSFP28/QSFP+ ports, dual hot-pluggable power supplies (AC/DC), six hot-swappable fan modules, console port and a 10M/100M/1G management port. ▪ Breakout panel support ▪ Console port (RJ45) and Management port (RJ45). ▪ In addition provides 2 x 10Gb/1Gb SFP+/SFP ports ▪ Cluster with GigaVUE HC Series nodes and GigaVUE-OS TA Series Nodes.	

GigaVUE-OS® HC Series Features and Benefits

Capable of port-to-port full line rate performance with minimal packet latency, the GigaVUE HC Series uses patented Flow Mapping® techniques to aggregate, replicate, and direct traffic flows, providing dynamic connectivity for 100Gb, 40Gb, 10Gb, or 1Gb monitor,

compliance, and archival tools, including:

- Intrusion Detection Systems
- Protocol Analyzers
- VoIP Analyzers
- Application Performance Monitors
- Stream-to-Disk Data Recorders

Any Packet, Any Destination

The GigaVUE HC Series nodes provide a powerful graphical user interface that lets you unobtrusively acquire and map traffic from multiple data sources to multiple tools, including the following common scenarios:

Mapping(Any-to-Any)	Direct traffic from any network port to any tool port. Use map rules to send different types of traffic to different tool ports.
Aggregation(Many-to-Any)	Aggregate traffic from multiple links to deliver a network-wide view to any tool. Merge Tx and Rx traffic into a single tool interface.
Multicasting(Any-to-Many)	Multicast filtered or unfiltered, singular or aggregated traffic to multiple tools.

The Gigamon Deep Observability Pipeline

GigaVUE-OS nodes and management software form the Gigamon Deep Observability Pipeline, providing passive monitoring of mission critical networks. The Deep Observability Pipeline solves access problems, improves network performance and uptime, and saves capital, operation and maintenance costs.

The Gigamon Deep Observability Pipeline addresses many common network management issues, including security, compliance, forensics review, application performance, and VoIP QoS, among others. Once data is acquired from multiple SPAN ports or TAPs, it can be multicast to multiple tools, aggregated to a few consolidated tools, and filtered or divided across many instances of the same tools.

You can think of the Deep Observability Pipeline as a data socket that provides immediate access for ad hoc tool deployment without impact to the production network. Gigamon's Deep Observability Pipeline nodes accommodate the growing number of network monitoring tools and network security tools. Figure 1Gigamon Deep Observability Pipeline summarizes these features.

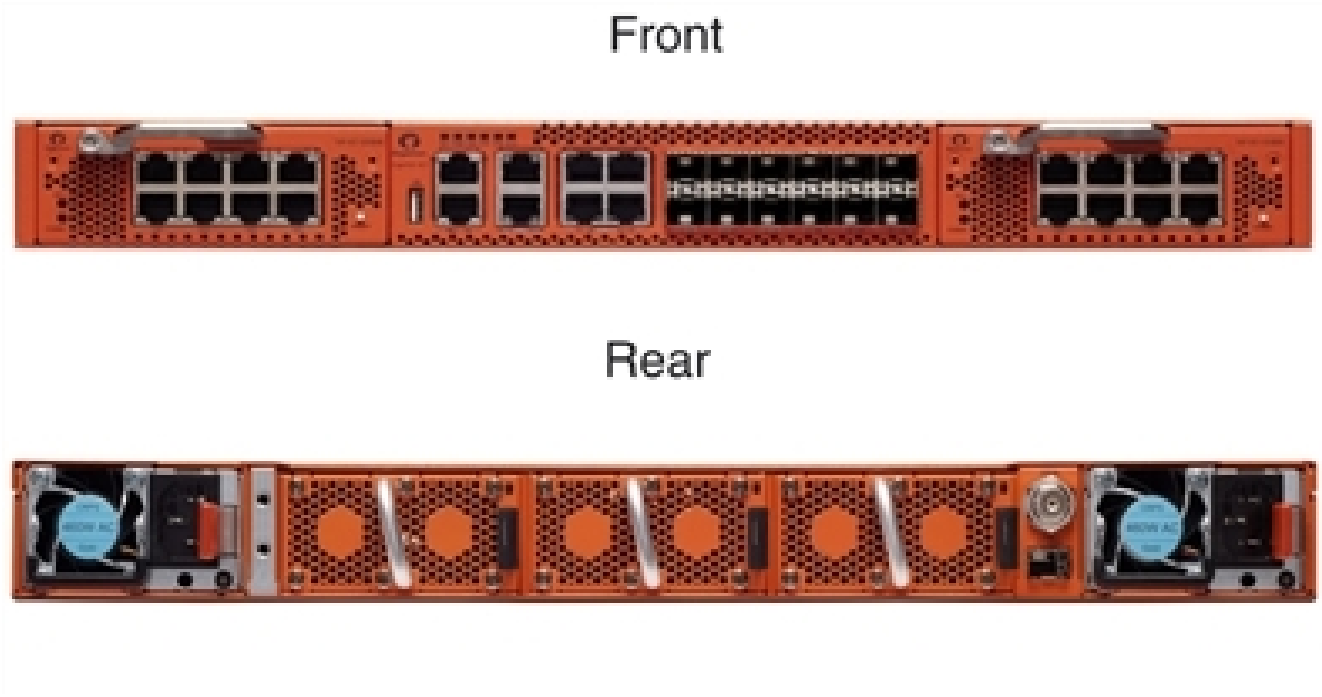


Figure 1 *Gigamon Deep Observability Pipeline*

GigaVUE-OS® HC Series Features and Benefits

The following table lists the major features and benefits of the GigaVUE-OS® HC Series:

Benefit	Descriptions
Web-Based Management	Manage the operations of theGigaVUE HC Series node using GigaVUE-FM, Gigamon's simple but powerful Web-based interface for GigaVUE HC Series nodes. GigaVUE-FM makes it easy to set up Flow Mapping, allowing you to see at a glance which network ports are delivering which packets to individual tool ports. Reconfigure Flow Mapping on the fly, selecting the packets you need when you need them.
CLI Management	Configure the operations of the GigaVUE HC Series node using a command-line interface, the GigaVUE-OS: • Local access over the serial **console** port on control card. • Remote network access using SSH2 over the 10/100/1000 Ethernet **Mgmt** port on control card. • Secure access to the CLI, either through local authentication or optional RADIUS/TACACS+/LDAP support.
Scalable Port Density	Use the line cards that best suit your port density needs. Depending on the line cards installed in the node, you can have as many as 256 10Gb ports (a node fully populated with PRT-H00-Q02X32 line cards). In addition, the GigaVUE HC Series node evolves with network speeds, including line cards with 40Gb and 100Gb support for data centers and service providers.

Benefit	Descriptions
Cluster Support	Connect multiple GigaVUE HC Series nodes in a self-healing, intelligent cluster. When you create a cluster of GigaVUE HC Series nodes, available ports appear as a unified fabric, with ingress ports able to send packets to any egress port, regardless of its physical chassis. Nodes are connected through stack links consisting of one or more 10Gb, 40Gb, or 100Gb ports. Cluster management traffic can be carried out-of-band on its own network or inband on stack links.
Share SPAN Ports	Connect a SPAN port to a network port on the GigaVUE HC Series node and multicast that traffic to multiple different tool ports, giving multiple different tools access to the same data. Use Flow Mapping to send specific traffic to different tool ports, ensuring that each tool sees the data that best suits its individual strengths. You can move, add, and reconfigure tools at will without affecting production networks.
Aggregate Links	Send the data from multiple different network ports to one or more tool ports, allowing you to combine traffic from multiple access points into a single stream for analysis.
Flow Mapping®	The GigaVUE HC Series Flow Mapping® features let you direct traffic arriving on network ports to one or more tool ports based on different packet criteria, including VLAN IDs, IP addresses, port ranges, protocols, bit patterns, and so on. You can drop some traffic intentionally using drop rules and also create a shared-collector destination for any packets not matching the maps configured on a shared set of network ports.
GigaVUE-FM Support	Deploy Gigamon's umbrella fabric management system, GigaVUE-FM to manage all of your GigaVUE HC Seriesnodes. The GigaVUE HC Series is fully compatible with GigaVUE-FM, allowing you to centralize deployment of images, configuration backups, and alert management.
Role-Based Access	Role-based access makes it easy to share the Gigamon Deep Observability Pipeline between different groups of users with different needs. Administrators can assign egress ports to different groups of users. Users can then select the traffic they need to see from shared ingress ports. Administrators adjust map priority to ensure that each packet is delivered to the correct destination.
Cisco-Style CLI	TheGigaVUE HC Series node's CLI offers a similar style to the familiar Cisco interface, minimizing relearning for IT professionals.
Command Abbreviation	Type only as many letters of a command as are needed to positively differentiate from other available commands. For example, you only need to type **co t** to enter Configure mode, not the full **configure terminal** command (although that works, too!).
SNMP Support	Rely on secure SNMP v3 access to the onboard SNMP agent as well as v1/v2 SNMP traps.
Email Notifications	Use email alerts for proactive notification of a wide variety of GigaVUE-OS events, helping you keep tabs on system status in real time.

Benefit	Descriptions
Modularized Design	Hot-pluggable line cards, power supplies, and fan trays allow for flexibility and future growth. For GigaVUE-HC1, and GigaVUE-HC3, the modules are interchangeable between the front bays of each chassis type, but not with each other, due to form and factor.
Flexible 10Gb/1Gb Support	All 10Gb ports in GigaVUE HC Series line cards can be used with 1Gb Ethernet media by inserting a copper or optical SX/LX SFP instead of an SFP+. Interoperability and support are ensured by purchasing SFPs from Gigamon – transceivers purchased from other vendors are not supported.
Flexible 100Gb/40Gb	All 100Gb ports in GigaVUE HC Series line cards can be used with 40Gb Ethernet media by inserting a QSFP+ instead of an QSFP28. Interoperability and support are ensured by purchasing SFPs from Gigamon – transceivers purchased from other vendors are not supported.

Module Numbering

Modules use standard conventions for numbering network and tool ports, both on the faceplates of the modules, as well as in the GigaVUE-OS CLI. On faceplates, the numbers are as follows:

100Gb Ports	Numbered with a leading **C**. For example, the SMT-HC3-C05 includes five 100Gb ports C1 to C5.
40Gb Ports	Numbered with a leading **Q**. For example, the PRT-HC0-Q06 includes 40Gb ports **Q1** to **Q6**.
25Gb/10Gb/1Gb Ports	Numbered with a leading **X**. For example, the PRT-HC3-X24 includes twenty-four 25Gb ports X1 to X24.
10/100/1000 Ports	Numbered with a leading **G**. For example, the PRT-HC1-X12G4 includes 10/100/1000 ports **G1** to **G4**.

The port labels on the module faceplates use upper-case C, Q, X, and G characters to identify ports. However, the CLI uses lowercase notation to refer to ports (for example, c1, q1, x4, and g1).

When referring to ports in the CLI, the format is **box ID/slot ID/port ID**. For example, **1/1/x6** refers to box 1, slot 1, port X6.

On chassis with multiple slots/bays, the slots or bays are numbered as follows:

- **GigaVUE-HC1**: Bays are numbered as follows:
  - the base chassis in the center, is numbered 1
  - the left module is numbered 2
  - the right module is numbered 3

- **GigaVUE-HC3**: Bays are numbered 1-4 from left upper, left lower, right upper to right lower.
- GigaVUE-TA series:

White Box Port and Faceplate Labeling

Unlike the GigaVUE-OS, the port number on a Certified Traffic Aggregation White Box (a white box) is a whole number, starting at one (1). CLI **show** commands display the faceplate numbering of all the ports on a white box chassis, as well as a mapping of the faceplate port number to the GigaVUE-OS port number and of the GigaVUE-OS port number to the faceplate port number.

Use the following CLI command to display faceplate numbering of all ports on a white box. Issue this command on a white box that is configured and whose operational status is *up*.

```
(config) show chassis box-id 3 faceplate-numbering
ONIE Faceplate Numbering:
+---+---+---+ +---+---+---+ +---+---+
| 1 | 3 | 5 | ... | 43| 45| 47| | 49| 51|
+---+---+---+ +---+---+---+ +---+---+
| 2 | 4 | 6 | ... | 44| 46| 48| | 50| 52|
+---+---+---+ +---+---+---+ +---+---+
GigaVUE-OS Faceplate Numbering:
+---+---+---+ +---+---+---+ +---+---+
| x1| x3| x5| ... |x43|x45|x47| | q1| q3|
+---+---+---+ +---+---+---+ +---+---+
| x2| x4| x6| ... |x44|x46|x48| | q2| q4|
+---+---+---+ +---+---+---+ +---+---+
Legend
------
x1..x48 : 10G ports
q1..q4 : 40G ports
```

Use the following CLI command to display the mapping of the faceplate port number to the GigaVUE-OS port number:

```
(config) # show port faceplate-number-mapping port-list 3/1/21,3/1/50..51,3/1/3..7
GigaVUE-OS Port Numbering: 3/1/x21,3/1/q2..q3,3/1/x3..x7
```

Use the following CLI command to display the mapping of the GigaVUE-OS port number to the faceplate port number:

```
(config) # show port port-number-mapping port-list 3/1/x1..x48,3/1/q1..q4
Faceplate Port Numbering: 3/1/1..48,3/1/49..52
```

When a card is not configured, the port mapping commands display an error message as follows:

```
(config) # show port faceplate-number-mapping port-list 3/1/1
Invalid port '3/1/1': invalid port syntax
```

When a card is not on a white box, the port mapping commands display an error message as follows:

```
(config) # show port port-number-mapping port-list 1/3/x1
% This command must be issued to a white box node.
```

For 40Gb ports that can be programmed to split to four SFP+ ports using an octopus cable or cable splitter, the subports are identified as follows:

```
1/1/48.1..48.2
```

This is equivalent to 1/1/x48..x49 on the GigaVUE TA Series.

For more information on white boxes and the Open Network Install Environment (ONIE), refer to *GigaVUE-OS Installation Guide on a White Box*.

Introducing the GigaVUE-OS CLI

This chapter introduces the GigaVUE HC Series command-line interface, the GigaVUE-OS, including basic techniques for entering commands and a summary of the available commands. Refer to the following sections for details:

- Command-Line Basics
- The Basic Commands
- What Is Saved In a Configuration File
- Saving a Configuration File
- Using the configuration Command
- Viewing the Contents of a Configuration File
- Applying Configuration Files
- Sharing Configuration Files with Other GigaVUE-OS® HC Series Nodes
- Module Numbering
- White Box Port and Faceplate Labeling

Accessing the Command-Line Interface

This chapter assumes you have already used the instructions in the **Hardware Installation Guide** for your node to unpack, assemble, rack mount, power on, and perform the initial configuration of the GigaVUE HC Series node in the GigaVUE-OS command-line interface.

Command-Line Basics

This section provides a quick orientation to the GigaVUE-OS command-line interface – how to enter commands, how to get help, and so on.

Command Line Modes

The GigaVUE-OS CLI can operate in one of three modes, each with its own set of available commands – Standard, Enable, and Configure. When you first launch the CLI, you start in Standard mode with access to a limited amount of commands used to review system status. As you move from **Standard** mode to **Enable** mode to **Configure** mode, both the power and the number of commands available increase, as summarized in Figure 1GigaVUE-OS Command-Line Modes Cheat Sheet.

Changing to Configure Mode

Change to **Configure** mode as follows:

1. Log in to the GigaVUE-OS. When you first log in, the CLI is in **Standard** mode, indicated by the **>** prompt (for example, ***[hostname]* >**)
2. Type **en <Enter>** to switch to Enable mode.

   The system prompt changes from ***[hostname]* >** to ***[hostname]* #**.
3. Type **config t <Enter>** to switch to Configure mode.

   The system prompt changes from ***[hostname]* #** to ***[hostname]* (config) #**.

If you are working over the serial console port, reset the terminal settings to match the current window with the **terminal resize** command.

Command Line Mode Cheat Sheet

The commands available in the CLI vary based on the mode (Standard, Enable, Configure) you are in and the user level (*admin* or *default*) of the logged in user. Figure 1GigaVUE-OS Command-Line Modes Cheat Sheetshows the different commands available in the each mode and to different levels of users.

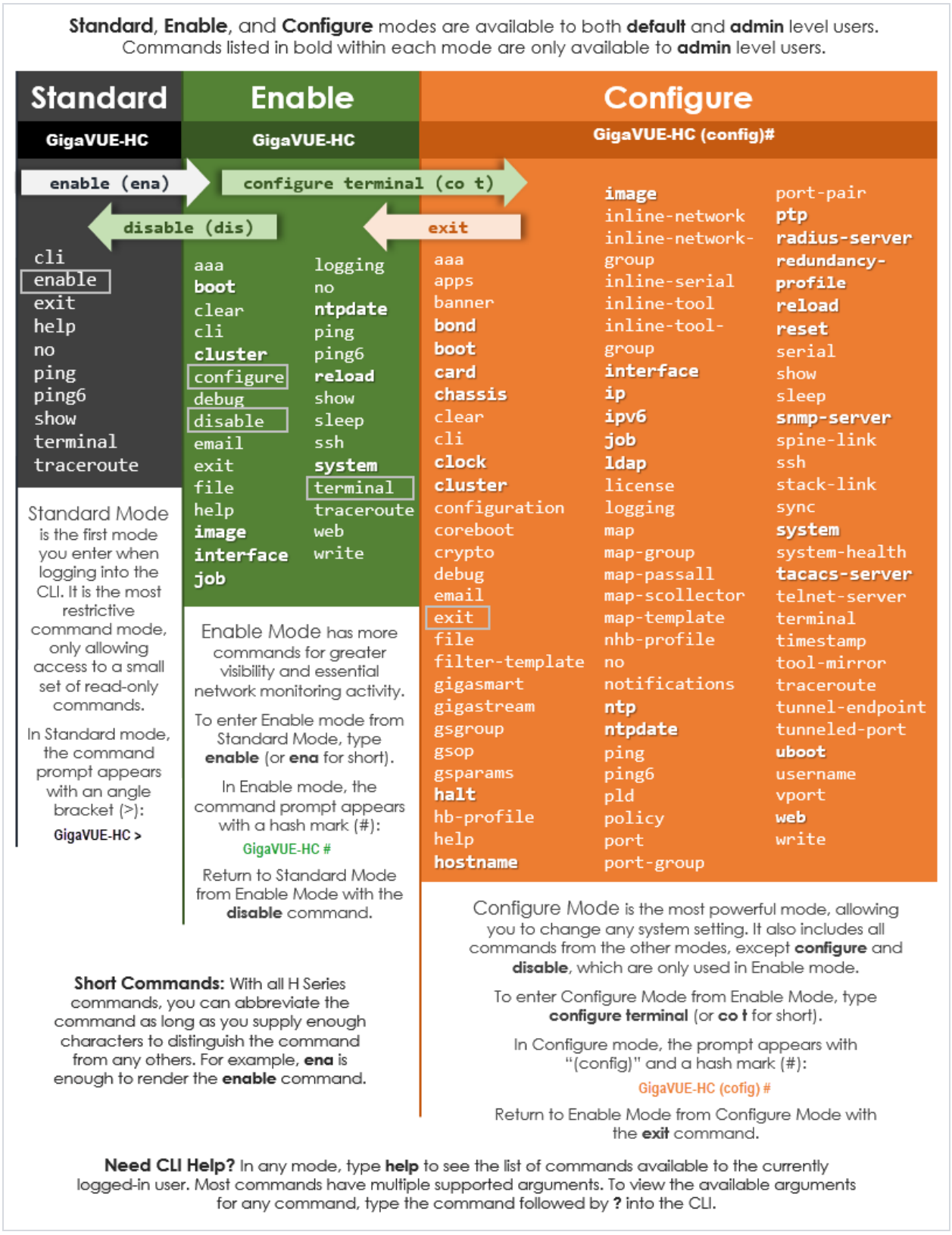


Figure 1 *GigaVUE-OS Command-Line Modes Cheat Sheet*

Changing Command Line Modes

The following table summarizes the commands used to change command-line modes:

Task	Command
Changing to Enable Mode You only need to supply enough of each command to positively identify it among the other available commands. So, in this example, you could just type **en** and press Enter.	**> enable**
Changing to Configure Mode Similarly, this command can be entered as **co t**.	**# configure terminal**
Notice how the system prompt changes with each command mode. Command modes offer greater control over the node as you ascend from **Standard** to **Enable** to **Configure**.	**(config) #**

Entering Commands in the CLI

The GigaVUE-OS provides several conventions that make it easy to identify available commands and enter them quickly:

Technique	Description
Context-Sensitive Help	The **?** symbol is the key to receiving help wherever you are in the CLI: ▪ Type the **?** by itself to see a list of all available commands. ▪ **Word Help** – Type a partial word with a **?** mark immediately following the partially-typed word to see a list of all possible commands using the word entered so far. For example, if you typed **r?** in Configure mode, the CLI would return the following possible commands based on what you have entered so far: **radius-server Configure RADIUS server settingsreload Reboot or shut down the systemreset Reset this system to its factory state** ▪ **Command Help** – Type a command followed by a question mark to see all possible arguments for the command as entered. If the system returns **<cr>**, that means the command can be entered as-is. For example, if you entered **gigastream ?**, you would see **alias**. You can build your way through the entire command tree by entering **?** after each new argument. For example, after entering **gigastream alias myalias ?**, you would see the next valid argument – **port-list**. **Note**: Typing **?** accesses the help system immediately – you do not need to press

Technique	Description
	<Enter>.
Partial Command Entry	For all GigaVUE HC Series commands and arguments, you only need to enter enough characters to distinguish the command from all other available commands. So, for example, you do not have to enter the full command username – there is only one command starting with the letter **d**, so you only have to enter **d**.
Command Completion	If you have partially typed a command, you can press **Tab** and the CLI will attempt to complete the command for you based on what has been entered so far. It is helpful to use the command completion feature together with partial command entry – you can press Tab while entering a command to see a list of all available commands matching what you have entered so far. For example, you can press **p<Tab>** and the system will return: **ping ping6 port port-group port-pair ptp** Based on this information, you know that you only need to enter the letter **t** to uniquely identify what you have entered as **ptp.**

Command-Line Syntax – Entering Commands

You enter CLI commands by typing enough characters to uniquely identify the command and pressing **<Enter>**.

When entering commands, keep in mind the following rules:

- Successful commands return no response in the CLI; commands with errors return an error response beginning with **%**, followed by a short error description.
- It is not recommended to use <CTRL+C> to cancel any CLI operation. In case, <CTRL+C> is issued on a running CLI command, the command runs in the background. When the background operation takes time to complete, exit from the CLI session and re-login to resume using the CLI.
- All commands are case-sensitive.
- Aliases are case-sensitive and accept both lower and upper case – for example, **map_alias** and **Map_Alias** refer to two separate maps.

  For a list of the special characters that cannot be used in aliases, refer to Alias Limitations.
- The **no** command is used to remove configuration settings. For example **no connect alias myconnect** deletes the named **myconnect**.
- Port numbers are entered in **<box ID>/<slot ID>/<port ID>** format. For example, **1/1/x1** identifies the 10Gb/1Gb port X1 in slot 1 on box 1 in the GigaVUE-OS. For details, refer to Module Numbering.
- Strings must consist entirely of alphanumeric characters with no spaces. The only exceptions are the underscore (_) and hyphen (-) characters. Those are allowed.

  For example, in Configure mode, **port 1/1/g1 alias Port_Alias** is legal, but **port 1/1/g1 alias Port Alias** is not.

> **NOTE:** Some string fields do accept spaces provided the input is made inside quotation marks (for example, the **banner login** command).

Configure Mode Syntax

Users of GigaVUE-OS nodes may be accustomed to entering the word **config** before many commands – **config map**, **config port-filter**, and so on. When using the GigaVUE-OS, the "config" part of the command is implied whenever you are working in Configure mode. The system prompt helps you remember this by including the word **(config)** in parentheses. For example:

```
(config) #
```

So, instead of entering **config gigastream** to set up a GigaStream consisting of multiple ports, you just enter **gigastream** followed by the necessary arguments. The **config** part is implied because you are already working in the Configure mode. For example:

```
(config)# gigastream alias mystream port-list 5/1/x1..x4
```

Paging Through CLI Output

By default, the CLI returns output exceeding the configured terminal length in pages. The CLI provides the same features for working through the paged output as the Linux programs **less** and **more**. Press the **h** key when presented with the paging prompt at the base of the display to see paging options – Figure 2Viewing Paging Options provides an example of how to do this.

If you prefer, you can disable paging either for the current session or by default:

- Disable for Current Session

  ```
  (config) # no cli session paging enable
  ```
- Disable for All Sessions (Default)

  ```
  (config) # no cli default paging enable
  ```

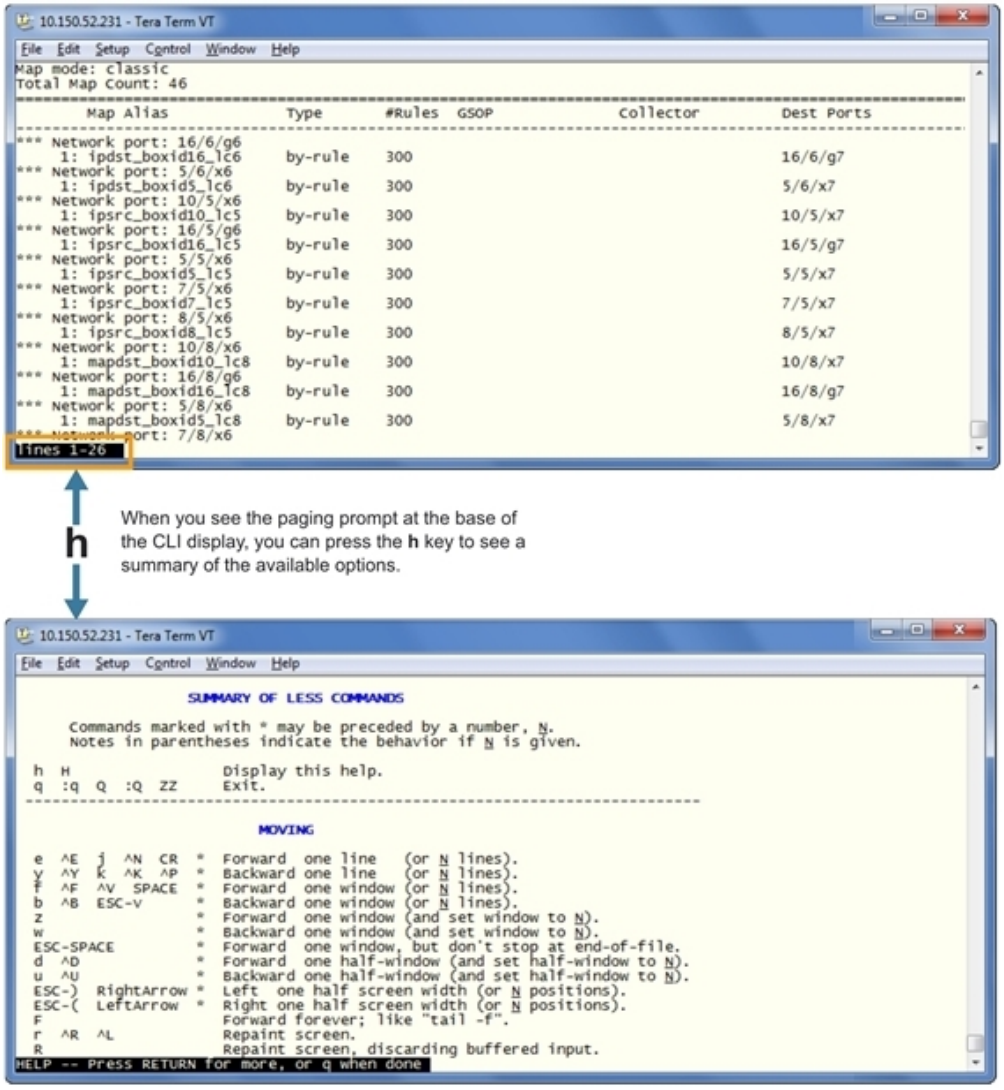


Figure 2 *Viewing Paging Options*

Tip: Assigning Unique Hostnames

If you are working simultaneously with multiple GigaVUE-OS nodes, you may want to assign each a separate hostname so its easy to identify separate terminal sessions from the system prompt. Admin users can do this with the **hostname <hostname>** command.

The Basic Commands

Refer to the Command Line Reference section for a list of commands for the GigaVUE-OS in Configure mode. As described in Figure 1GigaVUE-OS Command-Line Modes Cheat Sheet, the commands available in Configure mode are a superset of those available in Standard and Enable modes.

Most commands have multiple supported arguments. You can see the exact arguments available at any point of command entry by typing it into the CLI followed by **?**.

What Is Saved In a Configuration File

Configuration files store all of the settings in place on the GigaVUE-OS® HC Series node when the file was saved – everything necessary to restore the node to its exact state when the file was saved. This includes:

- Map settings
- Port aliases
- Port parameters, including duplex, medium, speed, cable length, and so on
- Port-groups
- Port-pair settings
- Tool-mirror settings
- Port-type settings
- GigaStream settings
- All settings shown by the **show system** command
- User accounts, groups, and roles
- SNMP server/trap settings
- TACACS+, RADIUS, and LDAP servers
- NTP servers
- Syslog servers
- Host names
- Mgmt port IP settings
- Logging settings, including email notifications

Saving a Configuration File

You can save the GigaVUE HC Series node's current systems to the active configuration file or a named file:

- Use the **write memory** command to save the running configuration to the current configuration file. Later on, when you start setting up packet distribution, your changes will be added to the active configuration right away, but will not be saved across a node reboot unless you use the **write memory** command to save your changes to flash.
- Use the **configuration write to** command to save the running configuration to a named configuration file. The named configuration file then becomes the active configuration file unless you include the **no-switch** argument at the end of the command. For example, this command writes the running configuration to a file named **config-bak** but leaves the current file active:

  ```
  (config) # configuration write to config-bak no-switch
  ```

NOTE: In contrast to the traditional GigaVUE-OS nodes, there is no requirement that GigaVUE HC Series Series configuration files have a **.cfg** extension.

Viewing Saved Configuration Files

Use the **show configuration files** command to see a list of available configuration files, as well as the currently active configuration file.

The name of the factory-provided configuration file is **initial**. You can see the name of the most recently booted configuration file by using the **show running-configuration** command (or **show configuration**) and look for the **## Running database** entry.

In Figure 3Showing the Current Configuration File

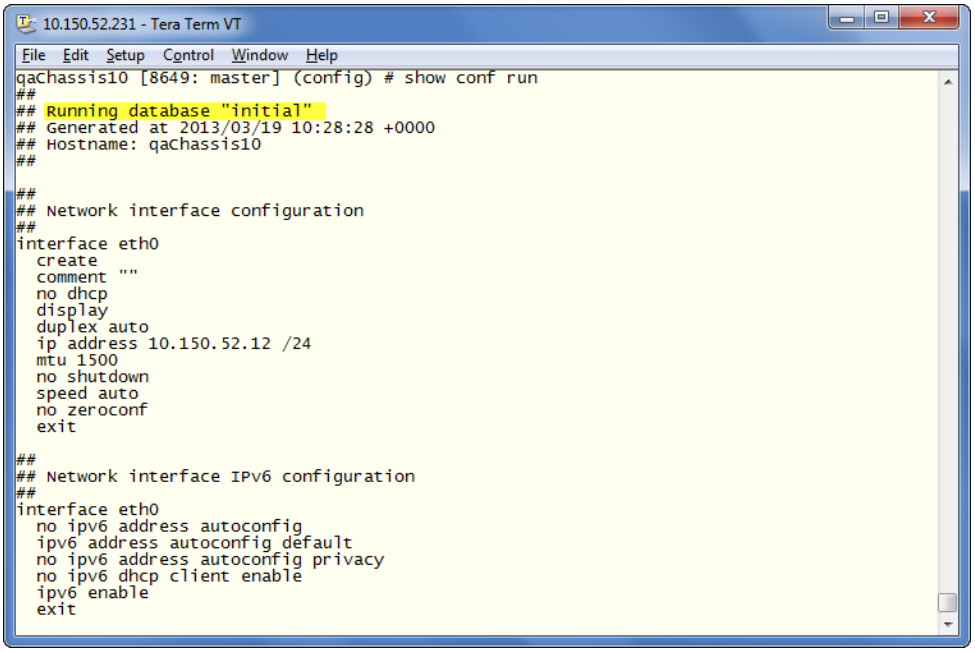


Figure 3 *Showing the Current Configuration File*

The file listed as **active** will load the next time the node is rebooted. For example, in Figure 4Showing Configuration Files:

- The **061013_config** configuration file is currently active and will load at the next boot – it is displayed with (active) after its entry.

NOTE: When you use the **show configuration files** command without a filename, you see the summary information shown in Figure 4Showing Configuration Files. You can also use the command with a filename to see detailed file information, as described in Viewing the Contents of a Configuration File.

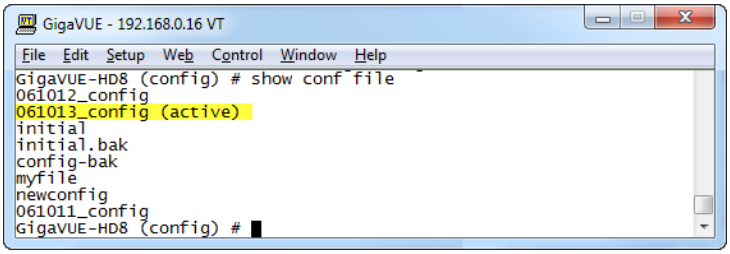


Figure 4 *Showing Configuration Files*

Using the configuration Command

Use the **configuration** command to manage configuration files on the GigaVUE HC Series node – separate arguments let you perform a wide variety of related tasks, including:

- Save, copy, and delete configuration files.
- Upload and retrieve configuration files from external hosts using FTP, TFTP, SCP, or SFTP.
- Show the contents of a configuration file.
- Load a saved configuration file.
- Return to a previous configuration file's settings.

Configuration File Types

There are two types of configuration files on the GigaVUE-OS® HC Series node – **standard** configuration files and **text** configuration files (known as **command files**):

- **Standard** configuration files can be used to store and apply a set of settings with the **configuration switch-to** command. Use this command to ensure that all configurations, including the order of map rule IDs are restored properly.
- **Text** configuration files are not really configuration files at all – instead, they are lists of CLI commands used to build a particular configuration. Text configuration files are useful for both troubleshooting and backup purposes – you can quickly see what commands built a particular configuration, or you can store regular backups of text files containing the commands on an external host. Text configuration files can also be applied in the CLI using the **configuration text file <filename> apply** command.

  You work with text configuration files using the **configuration text** command and its arguments.

Information Excluded from Text Configuration Files

For security reasons, text configuration files do not include plaintext passwords, such as SMTP passwords, AAA keys (RADIUS or TACACS+), private keys in RSA/DSA identities. Because of this, they cannot completely restore a given configuration using **configuration text file <filename> apply.**

Reserved Empty Database File

The empty database file, empty_db_file_dnu, is a reserved file. Do not use this filename (dnu) in any database operation such as **configuration write to** or **configuration switch-to** commands as the filename is removed when the node is reloaded.

Syntax for the configuration Command

For details on the **configuration** command, refer to configuration.

Viewing the Contents of a Configuration File

Restoring a configuration file to a GigaVUE-OS® HC Series node, overwrites the existing information in place on the node with the information stored in the configuration file. Because of this, check the contents of the file before you apply it.

You can easily see the details of what has been saved in a configuration file by using the **show configuration files [filename]** command:

- **show configuration files [filename]** displays the commands in the named configuration file.
- You can also use **write terminal** to display the commands necessary to recreate the current running configuration.

For example, to view the detailed contents of the **gigavue** file, you would use the following command:

```
show configuration files gigavue
```

Applying Configuration Files

Standard configuration files created on a node after cleaning the database can be applied only if the module configuration, such as line cards and transceivers, is exactly the same as the configuration file on that node that was backed up and saved before restarting.

This restriction does not apply for non-packet distribution commands, such as email or SNMP.

Apply configuration files to the GigaVUE-OS® HC Series node with the **configuration switch-to** command. The command applies the named configuration file, making it the active configuration file. So long as it remains the active configuration file, it will also be loaded the next time the node boots.

For example, the following command applies the configuration file named **gigavue**:

```
(config) # configuration switch-to gigavue
```

Sharing Configuration Files with Other GigaVUE-OS® HC Series Nodes

You can apply a configuration file created on one node to a second node. Keep in mind the following notes:

- All configuration settings that are not related to packet distribution (maps, tool-mirrors, port-pairs, and GigaStream) are reusable on the new node.
- Configuration settings related to packet distribution are tied to the chassis ID from the node on which they were saved. You can move these to the new node using either of the following methods:
  - Delete the old node (no chassis) and provision a new one, using a new box ID, if required.
  - If the box ID and module configuration of the new node is the same as the old node, you can perform a node migration using the procedure in the **Hardware Installation Guide**.

Command Line Reference

This section describes all GigaVUE-OS commands. The commands are organized alphabetically.

Refer to the top-level commands as follows:

Command	Description
aaa	Configures authentication, authorization, and accounting settings.
apps	Enables access to Gigamon Applications, such as Application Session Filtering (ASF), GTP backup, GTP whitelisting, Hardware Security Module (HSM), HSM group, icap, SSL Decryption for inline tools, keystore, NetFlow, SIP whitelisting, and Passive SSL decryption. GigaSMART applications are not supported on GigaVUE TA Series.
banner	Sets a system login banner.
bond	Configures bonded interface settings.
boot	Configures system boot parameters.
card (GigaVUE-OS® HC Series)	Configures a line card or module. On GigaVUE TA Series nodes, use the card (GigaVUE-OS®TA Series) command to enable additional ports.
chassis	Configures the GigaVUE HC Series chassis.
clear	Resets statistics or clears caches.
cli	Configures CLI shell options.
clock	Sets the system clock or timezone.
cluster	Configures a cluster of connected GigaVUE-OS nodes.
configuration	Manages configuration files.
configure	Enters Configure mode.
coreboot	Upgrades the BIOS image on GigaVUE-TA100, GigaVUE-HC1, and GigaVUE-HC3.
crypto	Manages X.509 certificates for theGigaVUE HC Series node's Web server.
debug	Generates a dump file for use in debugging issues with Gigamon Technical Support.
disable	Exits Enable mode and returns to Standard mode.
email	Configures email and event notification through email.
enable	Enters Enable mode.
exit	Exits Configure mode and returns to Enable mode or logs out of the CLI.
fabric advanced-hash	Configures advanced hashing parameters on stack GigaStreams and gsgroups.
file	Manages TCP and debug dump files on disk.
filter-template	Configures flexible filter templates on GigaVUE-HC3 and GigaVUE-OS-TA100.
gigasmart	Configures a stack port interface to provide Internet connectivity for a GigaSMART card or module.

Command	Description
gigastream	Configures a GigaStream—a group of ports acting as a single addressable tool port destination or stack-link.
gsgroup	Configures a GigaSMART group consisting of one or more GigaSMART engine ports. GigaSMART applications are not supported on GigaVUE TA Series.
gsop	Configures a GigaSMART operation consisting of one or more advanced processing applications. GigaSMART applications are not supported on GigaVUE TA Series.
gsparams	Configures GigaSMART parameters. GigaSMART applications are not supported on GigaVUE TA Series.
halt	Shuts down the system without powering it off.
hb-profile	Configures a heartbeat profile on GigaVUE HC Series nodes.
header-strip	Configure the Header Stripping protocol for a chassis.
help	Views a description of the interactive help system.
hostname	Specifies the system's hostname. The hostname appears in the system prompt and in SNMP traps.
ib-pathway	Configures the Resilient Inline Arrangement feature.
image	Manages system software images.
inline-network	Configures an inline network on GigaVUE HC Series nodes.
inline-network-group	Configures an inline network group onGigaVUE HC Series nodes.
inline-serial	Configures an inline tool series onGigaVUE HC Series nodes.
inline-tool	Configures an inline tool on GigaVUE HC Series nodes.
inline-tool-group	Configures an inline tool group on GigaVUE HC Series nodes.
interface	Configures network interfaces.
ip	Configures IP settings for the eth0 Mgmt port.
ip interface	Configures an IP interface to be used for GigaSMART encapsulation/decapsulation operations. This command is not supported on GigaVUE TA Series nodes.
ipv6	Configures IPv6 settings for the eth0 Mgmt port.
job	Configures scheduled jobs.
ldap	Configures LDAP server settings for authentication.
license	Activates features using license keys. Licensing is used for GigaSMART, Port, or Advanced Features License.
logging	Configures event logging.
map	Configures maps and map rules to manage GigaVUE-OS traffic distribution.
map-group	Configures map groups for GTP forward listing and GTP flow sampling.
map-passall	Creates a passall map to send all traffic on a network port to a tool port irrespective of the other packet distribution in place on the port.
map-scollector	Configures shared collector map parameters.
map-template	Creates a map template.
nhb-profile	Configures a negative heartbeat profile on GigaVUE HC Series nodes.

Command	Description
no	Deletes or clears certain configuration options.
notifications	Configures notification settings.
ntp	Enables and disables the use of NTP, as well as adds NTP servers.
ntpdate	Sets system clock once from a remote server using NTP.
onie	Reboots a Certified Traffic Aggregation White Box (white box) into Open Network Install Environment (ONIE) modes. This command is only available on white boxes where GigaVUE-OS is installed.
ping	Sends ICMP echo requests to a specified host.
ping6	Sends ICMPv6 echo requests to a specified host.
pld	Upgrades programmable logic devices (PLDs) on GigaVUE-HC3 nodes.
policy	Configures an Active Visibility policy.
port	Configures port type, parameters, and filters.
port-group	Creates a group of ports.
port-pair	Configures a port-pair on a pair of network ports within the same GigaVUE HC Series node. A port-pair is a bidirectional connection in which traffic arriving on one port in the pair is transmitted out the other (and vice-versa) as a pass through TAP.
ptp	Enables and disables the use of PTP.
radius-server	Configures RADIUS server settings for authentication.
redundancy-profile	Configures an inline redundancy profile onGigaVUE HC Series nodes.
reload (reboot)	Reboots or shuts down the node.
reset	Resets specified portions of the system configuration to their factory states.
serial	Sets options for the serial console port.
sfp	Reserved for future use.
show	Displays configuration and status information for GigaVUE HC Series settings and entities.
sleep	Sleeps for a specified number of seconds.
snmp-server	Configures SNMP settings, including the local SNMP server, notification events, and notification destinations.
spine-link	Configures spine links in a cluster with a leaf and spine architecture.
ssh	Enables and disables SSH access, as well as manages settings.
stack-link	Configures a stack-link between two GigaVUE HC Series nodes in a cluster. Stack-links are used to carry data traffic between nodes in a GigaVUE HC Series cluster.
system	Changes system settings.
system-health	Configures system health behaviors.
tacacs-server	Configures TACACS+ server settings for authentication.
terminal	Sets terminal parameters, including width and length.
tool-mirror	Configures a tool-mirror connection between two tool ports. A tool-mirror connection sends all packets arriving on one tool port to a second tool port on the same node.
traceroute	Traces the route packets take to a destination.

Command	Description
tunnel	Configures a circuit tunnel between two clusters.
tunnel-endpoint	Configures a tunnel endpoint, which is a destination for load balanced traffic from a L2GRE encapsulation tunnel.
uboot	Installs new uboot software.
username	Manages user accounts.
vport	Configures a virtual port. This command is not supported on GigaVUE TA Series nodes.
web	Configures the Web server used for the management of GigaVUE HC Series nodes using the GigaVUE-FM GUI.
write	Saves the running configuration to persistent storage.

General Information on Working with the CLI

Refer to Command-Line Basics for general instructions on working with the GigaVUE-OS.

Port Lists Definition in the GigaVUE-OS

Many CLI commands require that you specify a port list—port-filters, maps, and so on. The GigaVUE-OS observes a standard convention for port lists—you can use one or more of the following separated by commas—no spaces or tabs are allowed:

port-id <bid/sid/pid>

port-alias <port-alias>

port-list <bid/sid/pid_x..pid_y> (range) | <bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list) | gigastream-alias <gigastream-alias> | gigastream-alias-list | <gigastream-alias1,gigastream-alias2,...> | inline-network-alias <inline-network-alias> | inline-network-group-alias <inline-network-alias>

- The **port-list** argument lets you select multiple non-contiguous ports. To enter port IDs in a list, simply put a comma between each port ID in the list.
- The **<bid/sid/pid_x..pid_y>** argument lets you select a series of adjacent ports (for example, **1/5/x4..x6** selects port x4..x6 on slot 5.

  NOTE: Port ranges must be specified separately for 10Gb-capable and 1Gb ports. You cannot create a single range including both. For example, the PRT-H00-X12G04 card includes ports x1..x12 and ports g1..g4, but you cannot create a series that spans from **1/1/x1** to **1/1/g4**. Instead, you must create two series: **1/1/x1..x12** and **1/1/g1..g4**.

- GigaSMART load balancing port groups can have ports with different rates.
- You can mix a **port-id** with a **port-alias** and a **port-list** so long as they are separated by commas and no spaces. For example, **1/5/x4..x6,myalias,1/4/x2..x4** is a valid port-list.
- In some commands, the **port-list** includes a GigaStream alias, an inline-network alias, or an inline-network-group alias.

Examples

The port list conventions make it easy to connect multiple network ports or tool ports, for example:

Command	Comments
(config) # map-passall alias mymap (config map-passall alias mymap) # from 1/1/x1 **(config map-passall alias mymap) # to 1/4/x6..x8**	Creates a map that connects port 1/1/x1 to ports 1/4/x6, 1/4/x7, and 1/4/x8 with an alias of **mymap**.

Port Numbering/Speeds

The CLI uses lowercase notation to refer to ports (g1, x4, q1, c1 and d1). The port numbering refers to the following speeds:

- g—10M/100M/1000M ports
- x—10Gb/25Gb ports
- q—40Gb ports
- c—100Gb ports
- d—400Gb ports

Mode and User Level Commands

This section lists each command with an indication of the minimum command-line mode required for its use. Refer to and Figure 1GigaVUE-OS Command-Line Modes Cheat Sheet for a summary.

Most commands can be used by both admin and default-level users. Commands that can only be performed by admin-level users are listed as such.

aaa

Required Command-Line Mode = Enable

Use the **aaa** command to configure **accounting, authentication**, and **authorization** (AAA) settings for the GigaVUE-OS node.

Use the **aaa accounting** command to configure accounting settings. Refer to aaa accounting.

Use the **aaa authentication** command to configure authentication settings. Refer to aaa authentication.

Use the **aaa authorization** command to configure authorization settings. Refer to aaa authorization.

aaa accounting

Required Command-Line Mode = Configure

Use the **aaa accounting** command to configure accounting settings to enable or disable the logging of system changes to an AAA accounting server. Currently, TACACS+ is the only supported accounting server.

Configured TACACS+ servers are contacted in the order in which they appear in the configuration until one accepts the accounting data or the server list is exhausted.

The following table describes the arguments for the **aaa accounting** command:

Argument	Description
changes default stop-only <tacacs+>	Configures the order in which accounting changes default methods are tried as follows: • **stop-only**—Logs a TACACS+ accounting stop notification. • **tacacs+**—Specifies TACACS+ accounting method.

Related Commands

The following table summarizes other commands related to the **aaa accounting** command:

Task	Command
Displays general AAA settings.	**# show aaa**
Clears AAA accounting changes settings.	**(config) # no aaa accounting changes**
Clears the accounting changes default method list settings.	**(config) # no aaa accounting changes default**
Clears the accounting changes stop notification settings.	**(config) # no aaa accounting changes default stop-only**
Clears the accounting changes TACACS+ settings.	**(config) # no aaa accounting changes default stop-only tacacs+**

aaa authentication

Required Command-Line Mode = Configure

Use the **aaa authentication** command to specify the authentication methods to use for logins to the **Mgmt** port, as well as the order in which they should be used.

You can enable all authentication methods. If you enable more than one method, the GigaVUE-OS node uses the methods in the same order in which they are specified, falling back as necessary. If all servers using the first method are unreachable, the GigaVUE-OS node will fall back to the secondary method, and so on.

To prevent lockouts, it is recommended that you include **local** as one of the methods. However, the **local** method is optional. Refer to the "*Authentication*" section in the *GigaVUE Administration Guide*.

If a server responds to a login attempt with an authentication reject, no further servers using that method are tried. Instead, the next method is tried until either the user's login is granted or all specified methods are exhausted.

If you enable **radius**, **tacacs+**, or **ldap**, you must also:

- Configure the RADIUS, TACACS+, or LDAP server using the corresponding **radius-server**, **tacacs-server**, or **ldap** command.
- Configure GigaVUE-OS node users within the external authentication server itself.

The **aaa authentication** command has the following syntax:

```
aaa authentication
  attempts
    class-override
      admin no-lockout
      unknown <hash-username | no-track>
    lockout
      enable
      lock-time <seconds>
      max-fail <failure count>
      unlock-time <seconds>
    reset <all> | <user <username>> [no-clear-history | no-unlock]
    track enable
  certificate crl
    install name default pem url <URL>
    uninstall name default
  login default [ldap] [local] [radius] [tacacs+]
  password expiration
    duration <days>
    enable
```

The following table describes the arguments for the **aaa authentication** command:

Argument	Description
attempts class-override admin no-lockout	Overrides the global settings for tracking and lockouts for the admin account. Specifying **no-lockout** means that the admin user will never be locked out, though their authentication failure history will still be tracked if tracking is enabled overall. This applies only to the single account with the username admin. It does not apply to any other users with administrative privileges.
attempts class-override unknown <hash-username | no-track>	Overrides settings for the unknown class. Unknown means all usernames that are not recognized as real accounts (not a locally configured account). The overrides specify the following: • **no-track**—Does not track authentication for these users. • **hash-username**—Applies a hash function to the username and stores the hashed result.
attempts lockout enable	Enables or disables locking out of user accounts based on authentication failures. This suspends the enforcement of any existing lockouts and prevents any new lockouts from being recorded. If lockouts are later re-enabled, any lockouts that had been recorded previously, resume being enforced, but accounts that passed the **max-fail** limit are not automatically locked at this time. They are permitted one more attempt, and then locked out. Lockouts are applied after an authentication failure, if the user has surpassed the threshold at that time. Lockouts only work if tracking is enabled. Enabling lockouts will automatically enable tracking. Disabling tracking will automatically disable lockouts.
attempts lockout lock-time <seconds>	Specifies that no logins are permitted for this number of seconds following any login failure (not counting failures caused by the lockout mechanism, or the lock-time itself). This is not based on the number of consecutive failures. If both **unlock-time** and **lock-time** are set, the **unlock-time** must be greater than the **lock-time**. Unlike **max-fail**, this does take effect immediately for all accounts.
attempts lockout max-fail <failure count>	Sets the maximum number of consecutive authentication failures (attempts) permitted for a user account before the account is locked. After this number of failures, the account is locked and subsequent attempts are not permitted. This setting only impacts the lockouts imposed while the setting is active. It is not retroactive to previous logins. So if **max-fail** is disabled or changed, this does not immediately cause any users to be changed from locked to unlocked or vice-versa.
unlock-time <seconds>	Specifies that if a user account is locked due to authentication failures, another login attempt will be permitted if this number of seconds has elapsed since the last login failure. That does not count failures caused by the lockout mechanism itself. A user must have been permitted to attempt to login, and then failed. After this interval has elapsed, the account does not become unlocked, nor does its history reset. It simply permits one more login attempt even if the account is locked.

Argument	Description
	Unlike **max-fail**, this does take effect immediately for all accounts.
attempts reset <all> | <user <username>> [no-clear-history | no-unlock]	Clears the history of login failures, and/or unlocks the account. By default, both are done, which deletes the entire user record from the database. If either of the two optional parameters is used, the record is left in the database, but partially cleared. The parameters specify the following: • **no-clear-history**—Clears the history, but leaves the account's lock alone. Therefore, if it was locked, it remains locked until further action is taken. • **no-unlock**—Leaves the history alone and only unlocks the account. Therefore, one more login will be permitted, but the account could then become re-locked after another failure (if it was already over the threshold).
attempts track enable	Enables or disables tracking of authentication failures. The default is disabled. Tracking can be used for informational purposes or with the **lockout** argument. Disabling tracking does not clear any records of past authentication failures or the locks in the database. However, it prevents any updates to this database from being made. No new failures are recorded. It also disables lockout, preventing new lockouts from being recorded and existing lockouts from being enforced.
certificate crl install name default pem url <URL> uninstall name default	Configures certification authentication settings for Certificate Revocation List (CRL), as follows: • **install**—Downloads and installs a CRL as follows: o **name**—Specifies the name of the CRL to install. o **default**—Installs the specified CRL. o **pem**—Downloads and installs the specified CRL in PEM format. o **url**—Downloads the specified CRL in PEM format via the URL and installs it. • **uninstall**—Uninstalls a CRL as follows: o **name**—Specifies the name of the CRL to uninstall. o **default**—Uninstalls the specified CRL. Examples: **(config) # aaa authentication certificate crl install name default pem url http://192.168.1.2/godaddy.crl.pem** **(config) # aaa authentication certificate crl uninstall name default% NOTICE: local method is last in order and it will be used only if remoteservers are not reachable.**
login default [ldap] [local] [radius] [tacacs+]	Configures the order in which authentication methods for system logins are tried. The valid values are **ldap**, **local**, **radius**, and **tacacs+**. The order in which the methods are specified is the order in which the authentication is tried. To prevent lockouts, it is recommended that you include **local** as one of the methods. However, the **local** method is optional. In the following example, if local is not included as one of the methods, the device will be authenticated exclusively by the TACACS+ server:

Argument	Description
	(config) # aaa authentication login default tacacs+ Access is only given to one method at a time. In the following example, if the TACACS+ server is reachable, the local method will not be checked. Only if the TACACS+ server becomes unreachable will the method fall back to local. **(config) # aaa authentication login default tacacs+ local** In the following example, the local method will only be checked if neither the TACACS+ server or the RADIUS server are reachable: **(config) # aaa authentication login default tacacs+ radius local** In the following example, if the TACACS+ server is not reachable, the next method in order will be checked, which is local: **(config) # aaa authentication login default tacacs+ local radius**
password expiration duration <days> enable	Configures the number of days before a password expires and enables it for user accounts. When a user account is created, it is given the currently configured password expiration duration. If the duration is configured to 20 days, all user accounts that are created after that duration was configured, will expire after 20 days. If the duration is changed to 15 days, all user accounts that are created after that duration was configured, will expire after 15 days. That is, the user accounts configured to expire after 20 days, will not expire after 15 days as a result of the change to the duration. For example: **(config) # aaa authentication password expiration duration 20**

Related Commands

The following table summarizes other commands related to the **aaa authentication** command:

Task	Command
Displays general AAA settings.	**# show aaa**
Displays configuration and history of authentication failures.	**# show aaa authentication attempts**
Displays configuration of authentication failure tracking.	**# show aaa authentication attempts configured**
Displays status of authentication failure tracking and lockouts for all users.	**# show aaa authentication attempts status**
Displays failure tracking for a specified user.	**# show aaa authentication attempts status user manager**
Display authentication-certificate settings.	**# show aaa authentication certificate**
Displays the currently installed CRL.	**# show aaa authentication certificate crl name default**
Displays the tracked login attempts.	**# show aaa authentication login**

Task	Command
Displays the tracked login attempts for last x days (default 10). Add **last** to specify the number of days to track. Shows user name, # login attempts, hostname or IP address, and time. There are separate sections for successful and failed attempts.	**# show aaa authentication login** **# show aaa authentication login tracked last**
Deletes all overrides from the admin account.	**(config) # no aaa authentication attempts class-override admin**
Deletes the **no-lockout** override from the admin account.	**(config) # no aaa authentication attempts class-override admin no-lockout**
Deletes all overrides from unknown users.	**(config) # no aaa authentication attempts class-override unknown**
Deletes the **hash-username** override from unknown users.	**(config) # no aaa authentication attempts class-override unknown hash-username**
Deletes the **no-track** override from unknown users.	**(config) # no aaa authentication attempts class-override unknown no-track**
Disables lockout of accounts based on failed authentication attempts.	**(config) # no aaa authentication attempts lockout enable**
Disables temporary lock on account after every authentication failure.	**(config) # no aaa authentication attempts lockout lock-time**
Does not lock out users based on consecutive authentication failures.	**(config) # no aaa authentication attempts lockout max-fail**
Never allows authentication retry on locked account.	**(config) # no aaa authentication attempts lockout unlock-time**
Disables tracking of failed authentication attempts.	**(config) # no aaa authentication attempts track enable**
Clears authentication login settings.	**(config) # no aaa authentication login**
Negates authentication password expiration settings.	**(config) # no aaa authentication password expiration duration**
Disables password expirations.	**(config) # no aaa authentication password expiration enable**

aaa authorization

Required Command-Line Mode = Configure

Use the **aaa authorization** command to specify how externally logged-in users should be granted privileges on the GigaVUE-OS node. You can map all external logins to a specific local account, use matching accounts in the local database, or reject external logins unless they have a matching account in the local database.

The **aaa authorization** command has the following syntax:

```
aaa authorization
        map
                default-user <<user> | admin | monitor | operator
               order <<policy> | remote-only | remote-first | local-only>
        max-ssh-sessions <1-10>
        roles
                role <role name | Default> [description]
```

The following table describes the arguments for the **aaa authorization** command:

Argument	Description
map default-user <<user> | admin | monitor | operator>	Specifies the account to which externally authenticated logins are mapped when **map order** is set to **remote-first** (if there is no matching local account) or **local-only**.
map order <<policy> | remote-only | remote-first | local-only>	Specifies how externally authenticated logins (RADIUS, TACACS+, or LDAP) are mapped to local accounts, as follows: **remote-first**—Maps externally authenticated logins in the following order: **a.** Mapped to the matching local account name, if present. **b.** If there is no matching local account, the local user mapping attribute provided by the AAA server is used. **c.** If the local user mapping attribute is not present or does not specify a valid local user account, the account name specified by the **map default-user** argument is used. This is the default. **remote-only**—Maps externally authenticated logins in the following order: **a.** Mapped to the matching local account name, if present. **b.** If there is no matching local account, the local user mapping attribute provided by the AAA server is used. **c.** If the local user mapping attribute is not present or does not specify a valid local user account, no further mapping is attempted. **local-only**—Maps all externally authenticated logins to the user specified by the **aaa authorization map default-user <user name> command**.
max-ssh-sessions	Configure maximum concurrent SSH sessions. The session that you can configure ranges from 1 to 10.
role <role name | Default> [description]	Configures a role by name or Default and optionally adds a role description.

Related Commands

The following table summarizes other commands related to the **aaa authorization** command:

Task	Command
Displays general AAA settings.	**# show aaa**
Clears authorization user mapping default user settings.	**(config) # no aaa authorization map default-user**
Clears authorization user mapping order settings.	**(config) # no aaa authorization map order**
Reset maximum SSH sessions to default limit of 10	**(config) # no aaa authorization max-ssh-sessions**
Deletes a role definition.	**(config) # no aaa authorization roles role Default**
Deletes a description from a role.	**(config) # no aaa authorization roles role Default description**

apps

Required Command-Line Mode = Configure
Required User Level = Admin

Use the **apps** command to configure applications.

Use the **apps asf** command to configure Application Session Filtering (ASF) parameters. Refer to apps asf.

Use the **apps enhanced-lb** command to configure Enhanced Load Balancing (ELB) parameters. Refer to apps enhanced-lb.

Use the apps **enhanced-slicing** command to configure Enhanced Slicing Parameters. Refer to apps enhanced-slicing.

Use the **apps gtp-backup** command to manipulate GTP backup files. Refer to apps gtp-backup.

Use the **apps gtp-whitelist** command to configure GTP forward list parameters. Refer to apps gtp-whitelist.

Use the **apps hsm** command to configure Hardware Security Module (HSM). Refer to apps hsm.

Use the **apps hsm-group** command to configure an HSM group. Refer to apps hsm-group.

Use the **apps inline-ssl** command to configure inline Secure Sockets Layer (SSL) parameters for (Passive) SSL Decryption for inline tools. Refer to apps inline-ssl.

Use the **apps keystore** command to configure keystore key pairs. Refer to apps keystore.

Use the **apps netflow** command to configure NetFlow Generation parameters. Refer to apps netflow.

Use the **apps sip-whitelist** command to configure SIP forward list parameters. Refer to apps sip-whitelist.

Use the **apps ssl** command to configure Secure Sockets Layer (SSL) parameters for (Passive) SSL Decryption for out-of-band tools, or Passive SSL decryption. Refer to apps ssl.

apps asf

Use the **apps asf** command to configure Application Session Filtering (ASF) parameters. Use ASF after applying pattern matching with Adaptive Packet Filtering (APF). When a packet matches an APF rule, such as a regular expression filter rule, the subsequent packets with the same flow session will be forwarded to the same tool port as the matching packet.

Also use the **apps asf** command to configure ASF with buffering. Buffering ensures that all packets belonging to a flow session are captured and forwarded to the tools. For buffer ASF, you also need to allocate the number of session entries, in millions, using the **gsparams** command. Refer to **resource buffer-asf** under gsparams.

The **apps asf** command has the following syntax:

```
apps asf <alias <alias>>
  bi-directional <disable | enable>
  buffer <disable | enable>
  buffer-count-before-match <3-20>
  packet-count <2-100 | disable>
  protocol <tcp | udp | sctp | tcp-udp | tcp-udp-sctp>
  sess-field <add | delete>
    <gtpu-teid>
    <ipv4 | ipv4-5tuple | ipv4-dst | ipv4-l4port-dst | ipv4-protocol | ipv4-src | ipv4-src-l4port-
dst | ipv6 |  ipv6-5tuple | ipv6-dst | ipv6-l4port-dst | ipv6-protocol | ipv6-src | ipv6-src-l4port-
dst | l4port | l4portdst | l4portsrc> <inner | outer>
    <mpls-label | vlan-id> <pos <1 | 2>>
  timeout <10-120s>
```

The following table describes the arguments for the **apps asf** command:

Argument	Description
alias <alias>	Specifies the ASF alias. For example: **(config) # apps asf alias asf2**
bi-directional <disable | enable>	Specifies the direction of the flow, as follows: • **disable**—Disables capture of both directions of the flow. • **enable**—Enables capture of both directions of the flow. Depending on the session field attribute selected, GigaSMART will form the session field attribute for the reverse direction traffic. The default is **enable**, which means the opposite flow is captured. For example: **(config) # apps asf alias asf2 bi-directional disable** For details of bidirectional support, refer to Bidirectional Support for Session Field Attributes.
buffer <disable | enable>	Enables or disables buffer ASF. The default is **disable**. For example: **(config) # apps asf alias asf2 buffer enable** **Note**: To turn on buffer ASF, **buffer** must be enabled.
buffer-count-before-match <3-20>	Specifies the maximum number of packets that buffer ASF will buffer per session before an APF match. This provides a limit to the amount of buffering. The default is 3. The range is from 3 to 20. For example: **(config) # apps asf alias asf2 buffer-count-before-match 10**
packet-count <2-100 | disable>	Specifies the number of packets to forward to the tool port for each session match. After the packet count is reached, subsequent packets for the session are dropped. The packet count includes the packet that triggered the creation of the session. The default is **disable**, which means that all packets will be forwarded to the tool port. The range is from 2 to 100.

Argument	Description
	For example, to capture 50 packets after the pattern match: **(config) # apps asf alias asf2 packet-count 50** This parameter applies to APF pass rules (**gsrule add pass**). The number of packets dropped after the packet count is exceeded is displayed in the Exceed Count Drop field.
protocol <tcp | udp | sctp | tcp-udp | tcp-udp-sctp>	Specifies the protocol for buffer ASF as follows: • **tcp**—Specifies TCP only. • **udp**—Specifies UDP only. • **sctp**—Specifies SCTP only. • **tcp-udp**—Specifies both TCP and UDP. The default is **tcp**. • **tcp-udp-sctp**—Specifies TCP, UDP and SCTP. For example: **(config) # apps asf alias asf2 protocol udp**
sess-field <add | delete> <gtpu-teid> <ipv4 | ipv4-5tuple | ipv4-dst | ipv4-l4port-dst | ipv4-protocol | ipv4-src | ipv4-src-l4port-dst | ipv6 | ipv6-5tuple | ipv6-dst | ipv6-l4port-dst | ipv6-protocol | ipv6-src | ipv6-src-l4port-dst | l4port | l4portdst | l4portsrc> <inner | outer> <mpls-label | vlan-id> <pos <1 | 2>>	Specifies the attributes of a session field to add or delete. A session field is a group of fields that define a flow session. A flow session consists of one or more field names and attributes that define a session. Some field names include multiple attributes as follows: • **gtpu-teid**—GTP-u tunnel identifier. Not supported for buffer ASF. • **ipv4** (ipv4-src, ipv4-dst)—IPv4 source and destination IP. • **ipv4-5tuple** (ipv4-src, ipv4-dst, l4port-src, l4port-dst, ipv4-protocol)—IPv4 source and destination IP, Layer 4 (L4) source and destination port, and protocol field in IPv4 header. For buffer ASF, the IPv4 protocol is TCP/UDP. • **ipv4-dst**—IPv4 destination IP. • **ipv4-l4port-dst** (ipv4-src, ipv4-dst, l4port-dst)—IPv4 source and destination IP, and L4 destination port. • **ipv4-protocol**—Protocol field in IPv4 header. • **ipv4-src**—IPv4 source IP. • **ipv4-src-l4port-dst** (ipv4-src, l4port-dst)—IPv4 source IP and L4 destination port.

Argument	Description
	• **ipv6** (ipv6-src, ipv6-dst)—IPv6 source and destination IP. • **ipv6-5tuple** (ipv6-src, ipv6-dst, l4port-src, l4port-dst, ipv6-protocol)—IPv6 source and destination IP, L4 source and destination port, and protocol field in IPv6 header. For buffer ASF, the IPv6 protocol is TCP/UDP. • **ipv6-dst**—IPv6 destination IP. • **ipv6-l4port-dst** (ipv6-src, ipv6-dst, l4port-dst)—IPv6 source and destination IP, and L4 destination port. • **ipv6-protocol**—Protocol field in IPv6 header. • **ipv6-src**—IPv6 source IP. • **ipv6-src-l4port-dst** (ipv6-src, l4port-dst)—IPv6 source and L4 destination port. • **l4port** (l4port-src, l4port-dst)—L4 source and destination port. • **l4port-dst**—L4 destination port. • **l4port-src**—L4 source port. • **mpls-label**—MPLS label. • **vlan-id**—VLAN ID. In addition, for all IP and L4 port fields, specify the following: • **outer**—the first IP or L4 port in the packet. For buffer ASF, only **outer** is supported. • **inner**—the second IP or L4 port in the packet (usually inside tunneling). For MPLS label and VLAN ID fields only, position is the user-defined position of the field in the packet, as follows: **1**—the first occurrence of the protocol header or field in the packet. For buffer ASF, only position 1 is supported. **2**—the second occurrence of the protocol header or field in the packet. Examples: **(config) # apps asf alias asf1 sess-field add gtpu-teid** **(config) # apps asf alias asf2 sess-field add ipv4 inner**

Argument	Description
	(config) # apps asf alias asf3 sess-field add ipv4-5tuple oute) # apps asf alias asf4 sess-field add vlan-id pos 2
timeout <10-120s>	Specifies the session inactivity timeout, in seconds. A session will be removed due to inactivity when no packets match. The default is 15 seconds. The range is from 10 to 120 seconds. For example: **(config) # apps asf alias asf2 timeout 60**

Bidirectional Support for Session Field Attributes

The following table lists each session field attribute, the corresponding field for the reverse direction, and whether or not the bidirectional parameter is supported:

Field Attribute	Corresponding Field for Reverse Traffic	Bidirectional Support
ipv4-src	ipv4-dst	yes
ipv4-dst	ipv4-src	yes
ipv6-src	ipv6-dst	yes
ipv6-dst	ipv6-src	yes
l4port-src	l4port-dst	yes
l4port-dst	l4port-src	yes
ipv4-protocol	ipv4-protocol	yes
ipv6-protocol	ipv6-protocol	yes
vlan-id	vlan-id	yes
mpls-label	N/A	no
gtpu-teid	N/A	no

Related Commands

The following table summarizes other commands related to the **apps asf** command:

Task	Command
Displays configuration of a specified ASF.	**# show apps asf alias asf1**
Displays configuration of all ASFs.	**# show apps asf all**
Displays ASF statistics by alias.	**# show apps asf stats alias asf2**
Displays all ASF statistics.	**# show apps asf stats all**
Displays GSOP for ASF application.	**# show gsop by-application asf**
Displays GSOP statistics for ASF application.	**# show gsop stats by-application asf**
Deletes a specified ASF session field.	**(config) # apps asf alias asf2 sess-field delete gtpu-teid**
Deletes a specified ASF alias.	**(config) # no apps asf alias asf1**
Deletes all ASF aliases.	**(config) # no apps asf all**

apps gtp-backup

Use the **apps gtp-backup** command to manipulate GTP backup files created by GTP stateful session recovery.

The **apps gtp-backup** command has the following syntax:
(missing or bad snippet)

The following table describes the arguments for the **apps gtp-backup** command:

Argument	Description
delete <filename>	Deletes a GTP backup file by name. Use a question mark to display the names of backup files. The backup files have _backup and the following format: • **s2**—Specifies the slot number, for example, slot 2. • **e0**—Specifies the **e** port number, where 0 means **e** port 1 and 1 means **e** port 2. For example: **(config) # apps gtp-backup delete ?<filename>s2e0_backup** **(config) # apps gtp-backup delete s2e0_backup**
delete-all	Deletes all GTP backup files. For example: **(config) # apps gtp-backup delete-all**

apps exporter

Use the **apps exporter** command to configure the exporter.

> **NOTE:** The exporter configuration must be done on both the CPN and UPN to enable CPN UPN communication for filtering RAN and network slice attributes. If the exporter is not configured on both the CPN and UPN, the UPN will be in the non-CPN mode.

> **NOTE:** The exporter parameters, such as interface or port, cannot be modified for exporter type "mobility-cups". To modify the parameters of the exporter, delete the exporter and re-configure it.

> **NOTE:** The maximum number of exporters that can be configured for a given GigaSMART group in CPN UPN communication is 64.

The **apps exporter** command has the following syntax:

```
apps exporter alias <alias>

type <gtp-cups | tunnel | mobility-cups|netflow|tls-pcapng|mobility-sam>
description "<comment>"
source
   interface
   ip-interface <interface>
 l4
   port <1-65535>
destination
l3
   ip
   ver4 <IPv4-address>
   ver6 <IPv6-address>
   ttl <1-255>
   dscp <0-63>
   protocol <ipv4 | ipv6 | auto>
l4
  port <1-65535>
   protocol <tcp | udp>
ssl-profile <SSL-profile>
 tcp-profile <TCP-profile>
gsgroup <add | delete> <gsgroup_alias>
exit
```

The following table describes the arguments for the **apps exporter** command:

Argument	Description
alias <alias>	Specifies the exporter alias.
description	Add the description for the exporter profile.
type <gtp-cups | tunnel | mobility-cups|netflow|tls-pcapng|mobility-sam>	Specifies the type of the exporter as follows: • **gtp-cups** — GigaSMART host for GTP-Cups. • **tunnel**— GigaSMART host for Tunnel. • **mobility-cups**—GigaSMART host for CPN UPN. communication. • **NetFlow**—GigaSMART host for NetFlow. • **tls-pcapng**—GigaSMART host for TLS-PCAPng • **mobility-sam**—GigaSMART host for subscriber aware metadata.
source <interface <ip-interface> | l4 <port> <1-65535>>	Configures the following exporter source parameters: • **interface** — Configures the following exporter interface to be used. ▪ **ip interface** — Configure the exporter IP interface to be used • **l4** — Configures the following exporter source transportation layer parameters: ▪ **port**— Configure the exporter source port. **Note**: When the exporter type is **mobility-cups**, the UDP port must be within the ephemeral range (49152 to 65535).
destination <l3 | l4>	Configures the following exporter destination parameters: • **l3** — Configures the exporter destination network layer parameters. • **l4** — Configures the exporter destination transportation layer parameters. **NOTE:** When the listener type is **gtp-cups**, the configuration of exporter destination network layer parameters is optional.
l4< port <1-65535> protocol <tcp | udp>>	Configures the following transportation layer parameters in the exporter: • **port**— Configures the listener source port. The value ranges from 1 to 65535. • **protocol** — Configures the following listener l4 protocol o TCP o UDP
l3 <ip <ver6 <IPv6-address>| ver4 <IPv4-address>| ttl <1-255>dscp <0-63>> <protocol <ipv4 | ipv6 | auto>>	Configures the following network layer parameters in the exporter: • **dscp** — Configures the DSCP value. The value ranges from 0 to 63.

Argument	Description

	• **protocol** — Configures the following listener protocol.
	  ▪ IPv4 — Uses IPv4 for communication.
	  ▪ IPv6 — Uses IPv6 for communication.
	  ▪ Auto — Detects the protocol based on context.
	• **ttl** — Configures the Time-To-Live (TTL) in seconds. The value ranges from t to 255.

ssl-profile <SSL-profile>	Configures the SSL profile.
tcp-profile <TCP-profile>	Configures the TCP profile
gsgroup <add | delete>	Configures the gsgroup on which the exporter has to be activated or deactivated as follows: • **add** — Activates the exporter on which the gsgroup is configured. • **delete** —Deactivates the exporter on which the gsgroup is configured.

Related Commands

The following table summarizes other commands related to the **apps exporter**command:

Task	Command
Displays app exporter alias.	**#show apps exporter alias <alias>**
Displays statistics of apps exporters that has been configured. Given below is an example of apps exporter statistics with exporter type **mobility-cups**. This apps exporter statistics is not supported for exporter type **mobility-sam**. **HC3 (config) # show apps exporter stats all** **Total exporters Count: 128** **==========** **========== GigaSMART exporter Alias: en_cpn_exp1 ----------** **---------- Engine port: gsgrp-4g-cpn:1_4_e1** **Connections Initiated : 0** **Connections Received : 0** **Connections Success : 0** **Connections Failed : 0** **Connections Disconnected : 0** **Connection [20.0.0.2:60001 => 10.0.0.2:50001]** **Packets Received : 108** **Bytes Received : 5832**	**# show apps exporter stats all**

Task	Command
Packets Sent : 113 **Bytes Sent : 6102** **Packets Dropped : 0** **SFFP Sessions : 0** **GPFCP Tx HB : 113** **GPFCP Tx Association : 0** **GPFCP Tx Ran Update : 0** **GPFCP Tx Ran Resp Accept : 0** **GPFCP Tx Ran Resp Reject : 0** **GPFCP Tx Ran Pull Request : 0** **GPFCP Tx Ran Pull Req Retry : 0** **GPFCP Rx HB : 108** **GPFCP Rx Association : 0** **GPFCP Rx Ran Update : 0** **GPFCP Rx Ran Resp Accept : 0** **GPFCP Rx Ran Resp Reject : 0** **GPFCP Rx Ran Pull Request : 0** **GPFCP Session lookup fail : 0** **GPFCP Ack Missing : 0** **GPFCP RAN Retry : 0** **Connection Status : HB Active** **----------**	
Displays statistics of the specified apps exporter that has been configured.	**# show apps exporter stats <alias>**

apps metadata exporter

Use the **apps metadata exporter** command to configure the exporter.

The **apps metadata exporter**command has the following syntax:

```
apps-metata exporter alias
l4 port dest
l4 port src
max-pkt-size
source ip-interface attach
ttl
snmp enable/disable
```

```
type
exit
```

NOTE: Before you enable SNMP using **snmp enable** for your exporter ensure that Global SNMP is enabled using **snmp-server**. Refer to snmp-server.

Maximum Packet Size

The **max-pkt-size** parameter can be used to configure the maximum length of the packets that are exported from the AMI application. By default this option is disabled.

NOTE: The **max-pkt-size** parameter value can be configured between 1280 and 9400. When you configure the **max-pkt-size** value as 0, the feature will be disabled and the packets will get exported based on the available buffer size.

apps enhanced-slicing

Required Command-Line Mode = ConfigureRequired User Level = Admin

Use the **apps enhanced-slicing** command to configure Enhanced Slicing profile and their parameters.

The **apps enhanced-slicing** command has the following syntax:

apps enhanced-slicing alias <name>

protocol add <protocol fields> offset <offset-length> [flow-session <inner | outer> skip packet-count <1-50> [action <slice | drop>] [timeout <value>]]

protocol delete <rule-id>

max-sessions <max session entries>

exit

The following table describes the arguments for the **apps enhanced-slicing** command:

Argument	Description
enhanced-slicing alias <name>	Specifies an alias of the enhanced slicing file.
protocol add <protocol fields> offset <offset-length> [flow-session <inner | outer> skip packet-count <1-50> [action <slice | drop>] [timeout <value>]]	Adds the protocol to be added to the enhanced slicing profile. The protocols that can be added in the protocol field are as follows: • **ip**—Adds IP protocol. • **ipv4**—Adds IPv4 protocol. • **ipv6**—Adds IPv6 protocol. • **gtp**—Adds GTP protocol. • **gtp-ip**—Adds GTP-ip protocol. • **gtp-ipv4**—Adds GTP-ipv4 protocol. • **gtp-ipv6**—Adds GTP-ipv6 protocol.

Argument	Description
	• **tcp**—Adds TCP protocol. • **udp**—Adds UDP protocol. • **l4-port**—Adds layer 4 port number. The value of the port number ranges from 1 to 65535. • **gtpu-tcp**—Adds gtpu-tcp protocol. • **gtpu-udp**—Adds gtpu-udp protocol. For GTP protocol, the protocol position is inner by default. For other protocols, the protocol position must be configured either as inner or outer. At least one protocol field is required in each profile. The parameters that are configured are: • **offset <offset-length>** — Specifies the number of bytes that should be sliced after the protocol header. The offset-length value ranges from 64 to 9000 when there is no protocol selected. The value ranges from 0 to 9000 when other protocol is selected. • **flow-session** — Use flow session to slice the packet after it reaches a configured packet count from the beginning of a session flow. This is an optional attribute. Always use 4-tuple for IP and L4 port to identify a flow. The first occurrence of IP and L4 port is considered as the outer value. The second occurrence of IP and L4 is considered as the inner value. For a profile with flow-session defined, a session is created when a packet is received and when there is no existing flow-session for that flow. Slicing or dropping starts on the next packet of a session after the number of packets reaches the specified value in the packet count. **action** — Specifies to slice or drop packets after reaching the skip count (optional). The default action is slice. Slicing or drop is be performed in the following conditions: • When the traffic does not match any rule in the profile. • When a packet does not contain the configured inner or outer IP and l4-port in a defined flow-session. **timeout** — Specifies the session idle time (optional). The value ranges from 100 to 300. The default value is 30 seconds.
protocol delete <rule-id>	Deletes the protocol added to the enhanced silicing profile.
max-sessions <max session entries>	Specifies the maximum number of session entries, when the flow-session is configured. The value ranges from 4M to 80M. The default value is 4M.

apps enhanced asf

Use the **apps enhanced asf** command to configure the enhanced Application Session Filtering.

The **apps enhanced asf**command has the following syntax:

apps enhanced-asf
alias <name>
flow-session <outer | inner>
timeout <value in seconds>
max-sessions <max session entries>
rule add
transport <tcp | udp>
app <application protocol>
field <application field>
match-pattern <regex profile alias>
action <pass | drop>
rule delete <rule-id>
exit

The following table describes the arguments for the **apps exporter**command:

Argument	Description
alias <name>	Specifies the enhanced Application Session Filtering alias. Supports a maximum of 5 alias.
flow-session <outer | inner>	Inspect the location of the application (SSL/gQUIC/HTTP). Outer refers to the application protocol present right after first IP/L4 port. Inner refers to the application present after second (encapsulated) IP/L4 port.
timeout <value in seconds>	Specifies the session inactive time out. The value ranges from 10 to 300. The default value is 30 seconds.
max-sessions	Specifies the maximum number of session entries. The value ranges from 4 million to 10 million. The default value is 4 million
rule add	Configures the Filtering rule
transport <tcp | udp>	Specifies the either of the layer 4 protocol: • TCP • UDP
app <application protocol>	Specifies the following application protocol: • SSL • gQUIC • HTTP **Note**: IETF QUIC is not supported.
field <application field>	Specifies the application field: • SNI (SSL) • SNI (gQUIC) • HOST/User-agent (HTTP)

Argument	Description
match pattern <regrex alias name>	Specifies regrex pattern profile alias name.
action <pass | drop>	Forward or drops packets according to the match session.
rule delete <rule-id>	Deletes a configured rule.

Related Commands

The following table summarizes other commands related to the **apps enhanced asf**command:

Task	Command
Enables Enhanced Application Session Filteringfeature.	**gsop alias <gsop alias>** **apf set** **asf enhanced <enhanced asf alias>** **port-list <gsgroup name>**

apps enhanced-lb

Required Command-Line Mode = ConfigureRequired User Level = Admin

Use the **apps enhanced-lb** command to configure Enhanced load balancing profile and their parameters.

The **apps enhanced-lb** command has the following syntax:

apps enhanced-lb alias <elb-name>

hash-field <add | delete> <ip | ip-src | ip-dst> <inner | outer> [hash-mask <ipv4-mask> <ipv6-mask>] <l4-port | l4-port-src | l4-port-dst> <inner | outer> <gtpu-teid> hash-value add <ipv4-addr | ipv6-addr> <ip-addr> <mask> <inner | outer> [hash-mask <mask>]

hash-value delete <ipv4-addr | ipv6-addr> <rule-id>

exit

The following table describes the arguments for the **apps enhanced-lb** command:

Argument	Description
enhanced-lb alias <elb-name>	Specifies an alias of the enhanced load balancing file.
hash-field <add | delete> <ip | ip-src | ip-dst> <inner | outer> [hash-mask <ipv4-mask> <ipv6-mask>] <l4-port | l4-port-src |	Specifies the hash field information. • **ip-src**—Adds IP source address.

Argument	Description
l4-port-dst> <inner | outer> <gtpu-teid>	• **ip-dst**—Adds IP destination address. • **ipv4-mask**—Adds IPv4 mask. • **ipv6-mask**—Adds IPv6 mask. • **l4-port**—Adds layer 4 source and destination ports. • **gtpu-teid** — GPRS Tunnel Endpoint Identifier (TEID) configures the following field locations for hash: o **outer**—first occurrence of header or field o **inner**—second occurrence of header or field • **hash-mask**—Apply this mask before hashing IP for load balancing. The default value is 32 for ipv4 and 128 for ipv6. When hash-field ip-src or ip-dst is defined with ip, ip-src or ip-dst hash-mask overwrites ip hash mask. If field or fields in hash-field rules do not match incoming packets, packets are multi-casted to all the members in the output port group.
hash-value add <ipv4-addr | ipv6-addr> <ip-addr> <mask> <inner | outer> [hash-mask <mask>]	Adds the hash value. Supports up to 100 hash-value rules. If no hash-value rule is configured, field/fields in the hash-field rules is used for load balance hashing.
hash-value delete <ipv4-addr | ipv6-addr> <rule-id>	Deletes the hash value.

apps gtp-whitelist

Use the **apps gtp-whitelist** command to configure GTP forward listing.

NOTE: The **apps gtp-whitelist** commands are not persistent across a node restart, nor do they appear in the output of the running configuration.

The **apps gtp-whitelist** command has the following syntax:

apps gtp-whitelist alias <GTP whitelist file alias> add
imsi <IMSI number >
|ran <mcc.mnc.eci < eci number> | mcc.mnc.nci <nci number>>
<create | delete>
imsi <IMSI number> |
ran <mcc.mnc.eci < eci number> | mcc.mnc.nci <nci number>> | all
destroyfetch <add | delete> <URL for a GTP whitelist file>

The following table describes the arguments for the **apps gtp-whitelist** command:

Argument	Description
gtp-whitelist alias <GTP whitelist file alias>	Specifies an alias of the forward list file. Examples of valid names are wlist, imsi-database_2.
add imsi <IMSI number| ran <mcc.mnc.eci < eci number> | mcc.mnc.nci <nci number>>>	Specifies actions for **add** as follows: • **imsi**—adds a single IMSI to a forward list. • **ran**—adds RAN value to add **mcc.mnc.eci** or **mcc.mnc.nci** value. The value should be given in the following format: o NCI value in hexadecimal format and should add 0x as prefix. o ECI value in decimal format and supports up to 9 digits. For example: **(config) # apps gtp-whitelist alias wlf1 add imsi 318260109318283** **(config) # apps gtp-whitelist alias ran_db add ran 210.32.345678912** **apps gtp-whitelist alias ran_db add ran 755.56.0xf12345678**
create	Creates a new forward list. For example: **(config) # apps gtp-whitelist alias wlf1 create** To create a whitelist, refer to How to Create a Forward List.
delete <all | imsi <IMSI number> |ran <mcc.mnc.eci < eci number> | mcc.mnc.nci <nci number>>>	Specifies actions for **delete** as follows: • **all**—Deletes a forward list. This deletes all the IMSI and RAN entries. • **imsi**—Deletes a single IMSI entry from a forward list. When using **delete all** to delete a forward list, unlike **destroy**, you do not have to delete the forward list maps, the GigaSMART operation, or disassociate the GigaSMART group from the forward list. • **ran**—adds RAN value to add **mcc.mnc.eci** or **mcc.mnc.nci** value. The value should be given in the following format: o NCI value in hexadecimal format and should add 0x as prefix. o ECI value in decimal format and supports up to 9 digits. Examples: **(config) # apps gtp-whitelist alias wlf1 delete imsi 318260109318283** **(config) # apps gtp-whitelist alias wlf1 delete all**
destroy	Destroys a forward list. For example: **(config) # apps gtp-whitelist alias wlf1 destroy** When using **destroy** to delete a forward list, unlike **delete all**, you must first delete the forward list maps, the GigaSMART operation, and disassociate the GigaSMART group from the forward list before deleting the forward list. For the procedure to destroy the forward list, refer to How to Delete a Forward List.

Argument	Description
fetch <add | delete> <URL for a GTP whitelist file>	Specifies actions for **fetch** as follows: • **add**—Downloads a forward list file from a specified URL and path. • **delete**—Deletes the IMSI and RAN entries, located in the forward list file at the specified URL and path, from the forward list on the node. Use this option to delete up to 50,000 IMSIs. For both **add** and **delete**, forward list files must adhere to the following: • The IMSIs or RAN entries in the forward list files must be distinct entries, with one IMSI or RAN on each line of a file. • In a forward list file, use only the carriage return (newline) to separate IMSI or RAN entries. Do not use any characters, such as commas or colons, to separate IMSI or RAN entries in forward list files. • Each forward list file can contain a maximum of 50,000 entries. • Forward list files must have a filename with a .txt suffix. To fetch a specified forward list file from a location, use one of the following formats: • http://IPaddress/path/filename.txt • scp://username:password@IPaddress:/path/filename.txt For GTP forward listing in a cluster, only fetch the forward list to the leader in the cluster. On member nodes, **fetch** is not available. Examples: **(config) # apps gtp-whitelist alias wlf1 fetch add http://1.1.1.1/tftp/temp/MyIMSIs1.txt** **(config) # apps gtp-whitelist alias wlf2 fetch add scp://user1:mypw@1.1.1.1:/home/temp/IMSI_file1.txt** **(config) # apps gtp-whitelist alias wlf3 fetch add tftp://192.168.51.41/temp/IMSI_20K_1.txt** **(config) # apps gtp-whitelist alias wlf1 fetch delete http://1.1.1.1/tftp/temp/MyIMSIstoDelete.txt** **(config) # apps gtp-whitelist alias wlf2 fetch delete scp://user1:mypw@1.1.1.1:/home/temp/IMSI_delfile.txt** **Note**: In a single forward list file, both IMSI and RAN entries are supported. RAN entries should be given in the format as specified in the **add** option for single entry.

How to Create a Forward List

To create a forward list, use the following CLI command sequence:

Task	Command
Create the forward list.	**(config) # apps gtp-whitelist alias wlf1 create**
Associate the GigaSMART group to the forward	**(config) # gsparams gsgroup gsg1 gtp-whitelist add**

Task	Command
list.	**wlf1**
Configure the GigaSMART operation.	**(config) #** gsop alias **gtp_wl1** flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg1
Add single entry to the whitelist.orFetch and download forward list files.	**(config) # apps gtp-whitelist alias wlf1 add imsi 318260109318283** **(config) # apps gtp-whitelist alias wlf1 add imsi 318573850131409** **(config) # apps gtp-whitelist alias wlf1 fetch add http://1.1.1.1/tftp/temp/whitelist1.txt** **(config) # apps gtp-whitelist alias wlf1 fetch add http://1.1.1.1/tftp/temp/whitelist2.txt**
Create from one to ten second level maps, the forward list maps. When the map configuration is complete, the forward list will take effect. **Note**: If no whitelist add rule is specified in the map, all traffic (all interfaces and all versions) will be passed.	**(config) # map alias GTP-Whitelist** **(config map alias GTP-Whitelist) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist) # from vp1(** **config map alias GTP-Whitelist) # use gsop gtp_wl1** **(config map alias GTP-Whitelist) # to 1/2/x2** **(config map alias GTP-Whitelist) # whitelist add gtp version 2** **(config map alias GTP-Whitelist) # exit** **(config) #**

How to Delete a Forward List

To destroy the entire forward list, use the following CLI command sequence:

Task	Command
Delete a forward list map.	**(config) # no map alias GTP-Whitelist**
Delete the GigaSMART operation.	**(config) # no gsop alias gtp_wl1**
Disassociate the GigaSMART group from the forward list. (You need to delete and re-add the **gsgroup**.)	**(config) # gsparams gsgroup gsg1 gtp-whitelist delete**
Destroy () the entire forward list.	**(config) # apps gtp-whitelist alias wlf1 destroy**

Related Commands

The following table summarizes other commands related to the **apps gtp-whitelist** command:

Task	Command
Configures a rule for a forward list map.	**# map alias <whitelist map> whitelist add gtp <interface | version>**
Displays a particular IMSI associated with the GigaSMART group.	**# show gsgroup flow-whitelist alias gsg1 imsi 318260109318283**
Displays the GTP forward list entry count.	**# show apps gtp-whitelist alias wlf1 count**
For forward list maps, displays the total number of IMSI entries (under WL).	**# show map brief**
For forward list maps, displays the total number of IMSI entries.	**# show map alias <whitelist map>**
For forward list maps, displays the total number of IMSI entries.	**# show map stats alias <whitelist map>**

apps hsm

> **NOTE:** Starting in release 6.14.00, Entrust/nCipher HSM integration is deprecated. Gen2 DSSL HSMs (nCipher v1.0) and Gen3 HSMs using nCipher 12.6 (`nshield_12_60`) are no longer supported, and upgrades are blocked if any nCipher HSM configuration is present.

Use the **apps hsm** command to configure a Hardware Security Module (HSM) appliance.

The **apps hsm** command has the following syntax:

apps hsm <alias <alias>>

hsm-ip <HSM server IP address> hsm-port <port number> type luna-hsm server-username <name> server-password <***> partition-label <name> partition-password <*****>**

The following table describes the arguments for the **apps hsm** command:

Argument	Description
alias <alias> **hsm-ip <HSM server IP address> hsm-port <port number> type luna-hsm server-username <name> server-password <*****> partition-label <name> partition-password <*****>**	Configures an HSM appliance as follows • **alias**—Specifies an alias of the HSM. • **hsm-ip**—Specifies the IP address of the HSM server. Only IPv4 addresses are supported. • **hsm-port**—Specifies the HSM port number. • **type luna-hsm**—Specifies the HSM type which is Thales-Luna. • **server-username**— Specifies the HSM servers

Argument	Description
	administration username. • **server-password**— Specifies the HSM servers administration password. • **partition label**—Specifies the user partition label configured by the administrator. • **partition password**—Specifies the user partition password configured by the administrator. Examples: **(config) # apps hsm alias hsm1 hsm-ip 10.115.72.15 hsm-port 1792 type luna-hsm server-username admin server-password ******* partition-label partition1 partition-password ******* **(config) # apps hsm alias hsm2 hsm-ip 10.115.74.36 hsm-port 1792 type luna-hsm server-username admin server-password ******* partition-label partition1 partition-password ******* **NOTE:** The server-password and partition-password should be encrypted using the keychain password in the keystore.

Related Commands

The following table summarizes other commands related to the **apps hsm** command:

Task	Command
Displays a specified HSM.	**# show apps hsm alias hsm1**
Displays all HSM.	**# show apps hsm all**
Deletes a specified HSM.	**(config) # no apps hsm alias hsm1**
Deletes all HSM.	**(config) # no apps hsm all**

apps hsm-group

NOTE: Starting in release 6.14.00, Entrust/nCipher HSM integration is deprecated. Gen2 DSSL HSMs (nCipher v1.0) and Gen3 HSMs using nCipher 12.6 (`nshield_12_60`) are no longer supported, and upgrades are blocked if any nCipher HSM configuration is present.

Use the **apps hsm-group** command to configure an HSM group.

The **apps hsm-group** command has the following syntax:

```
apps hsm-group <alias <alias>>
type luna-hsm
    comm <comment>

    hsm-alias
            add <HSM alias>
            delete <HSM alias>
```

The following table describes the arguments for the **apps hsm-group** command:

Argument	Description
alias <alias>	Specifies an alias of the HSM group. For example: **(config) # apps hsm-group alias hsm-set** **Note**: Only one HSM group can be configured.
type luna-hsm	Specifies the vendor type Thales-Luna HSM.
comment <comment>	Add a comment to an HSM group. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # apps hsm-group alias hsm-set comment "HSM group1"**
hsm-alias add <HSM alias> delete <HSM alias>	Specifies the HSM alias to add or delete as follows: • **add**—Adds an HSM to an HSM group. Multiple HSMs can be added to a group. Multiple HSMs might be needed for load balancing, failover, or redundancy. • **delete**—Deletes an HSM from an HSM group. Examples: **(config) # apps hsm-group alias hsm-set hsm-alias add hsm1** **(config) # apps hsm-group alias hsm-set hsm-alias add hsm2** **(config) # apps hsm-group alias hsm-set hsm-alias delete hsm1**
hsm-set fetch keymap <URL>	Fetches the text file with the key mappings from the specified URL. You must create a text file with the key mappings and upload it to a server. Enter a valid directory path including the text file name. It is recommended to use a secure protocol, such as SCP or HTTPS to access the URL. Example of the Keymap text file format: **server-ip key-name/key-token** 20.1.1.1 rsa2048-server1-cert 20.1.1.2 key_pkcs11_uad6963c0f0c30037c707e22ed6ccf8e12014a237d 20.1.1.3 433 rsa2048-server1-cert 20.1.1.4 433 key_pkcs11_uad6963c0f0c30037c707e22ed6ccf8e12014a237d Example:

Argument	Description
	(config) # apps hsm-group alias hsm-set fetch keymap scp://user@10.10.10.10/keymap.txt

Related Commands

The following table summarizes other commands related to the **apps hsm-group** command:

Task	Command
Displays Security World information.	# show apps hsm-group world
Displays SSL session statistics that are handled by HSM servers.	# show apps hsm-group session-stats
Displays HSM buffer statistics.	# show apps hsm-group buffer-stats
Displays all statistics along with their operation state.	# show apps hsm-group all
Displays operational status along with their operation state. The operational states are as follows: • **RegisterInProgress**-This state indicates that the registration process of HSM-LUNA client on HSM-LUNA server is in progress. • **RegisterPending** -The register-pending state indicates that the Luna client registration on HSM-LUNA server is pending. **Note**: To register your clinet in Luna server, refer to Client Register - Luna Command Reference for more details. • **VerifyInProgress** - This states indicates That the verification has begun after LUNA client registration. • **HAConfig**- This state denotes that the prerequisite configurations are completed for HA verification and HA verifications has started/in-progress.YES means hsm configuration is correct and up. • **YES** - This state denotes that the HA verification is completed successfully and the HSM configuration is functional. • **NO** - The HSM configuration is not functional. • **INIT** - This state denotes that the HSM Server configuration has been initialized in	# show apps hsm-group status

Task	Command
GigaSMART node. This also checks the communication between GigaSMART node and HSM server. • **Unknown**- The HSM Group was configured prior to the Inline SSL or Passive SSL Application was configured.	
Displays the list of key labels available in Thales-Luna HSM group.	#show apps hsm-group keylabels
Deletes a specified HSM group.	(config) # no apps hsm-group alias hsm-set
Deletes all HSM groups.	(config) # no apps hsm-group all
Verifies if the Luna Network HSM slots/partitions are visible to the Client.	#show apps hsm-group verify
Verifies if the Luna HSM appliances are pingable.	# show apps hsm-group ping-result
Verifies the Luna HSM appliances HA statistics.	# show apps hsm-group ha
Displays HSM server information	# show apps hsm-group server

apps icap

Use the **apps icap** command to configure the ICAP Client app.

The **apps icap** command has the following syntax:

apps icap
server alias <alias>
server-group alias <group>
profile alias <icap-profile-alias>
gsop alias <gsop-alias> icap <icap-profile-alias> port-list <port-list>

The following table describes the arguments for the **icap** command.

Argument	Description
server alias	Specifies the name of the icap server. ICAP server can be IPV4 or IPV6 config based server. A maximum of 10 ICAP servers can be created in the system.
server-group alias	Specifies the icap server group. A maximum of 10 ICAP servers can be configured in an ICAP group.

Argument	Description
profile alias	Specifies the name of the icap profile and describes the ICAP Client profile configuration.
port-list <port-list>	Specifies engine ports. Use one or more of the following separated by commas—no spaces or tabs are allowed: • port-id—<bid/sid/pid> where pid is e1 or e2 • port-list—<bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list) where pid is e1 or e2 The port-list argument lets you select multiple non-contiguous ports. To enter port IDs in a list, put a comma between each port ID in the list.

Related Commands

The following table summarizes other commands related to the **apps icap**command:

Task	Command
Displays all icap server.	**# show apps icap server all | <alias>**
Displays all icap server groups.	**# show apps icap server-group all | <alias>**
Displays all icap profiles.	**# show apps icap profile all | <alias>**
Displays GSOP for icap.	**# show gsop by-application icap**
Displays icap server stats.	**# show icap stats server alias <server-alias>**
Displays icap session summary information.	**# show apps icap session summary**
Displays active connections being intercepted by ICAP server.	**# show apps icap session any**

apps inline-ssl

Use the **apps inline-ssl** command to configure inline Secure Sockets Layer (SSL) parameters for SSL Decryption for inline tools. For more information, refer to the "*Work With Inline SSL Decryption"* section in the *GigaVUE Fabric Management Guide*.

The **apps inline-ssl** command has the following syntax:

```
apps inline-ssl
  caching persistence <disable | enable>
  keychain password <password> <confirm password> | <password> | [reset]
<password><confirm password>
version < above | below >
  min-version <sslv3 | tls1 | tls11 | tls12 | tls13> max-version <sslv3 | tls1 | tls11 | tls12 | tls13>
below min-version <no-decrypt | drop>
above max-version <no-decrypt | drop >
  profile alias <alias>
```

inbound tool-early-inspect <enable | disable>

inbound tool-early-inspect connection timeout <1-10 sec>

split-proxy [enable | disable]

server non-pfs-ciphers [enable | disable]

tool early-engage [enable | disable]

one-arm [enable | disable]

monitor <disable | enable | inline>
certificate
expired < decrypt | drop>
invalid <decrypt | drop>
revocation crl <disable | enable [fail <hard | soft>] [defer timeout <20-100>]>
revocation ocsp <disable | enable [fail <hard | soft>] [defer timeout <20-100>]>
self-signed <decrypt | drop>
unknown-ca <decrypt | drop>

client-auth trust-store [add <trust-store-alias> | delete]

client-auth primary-signing [<key_alias> | delete]

client-auth secondary-signing [<key_alias> | delete]

client-auth revocation [crl | ocsp] [enable | disable] [fail <hard | soft>] [defer timeout <20–100>]

client-auth unknow-ca <decrypt | drop> client-auth expired <decrypt | drop> client-auth invalid <decrypt | drop> client-auth self-signed <decrypt | drop>

clear<decryptlist | nodecryptlist>

decrypt
tcp
inactive-timeout <2-1440 mins>
timewait-timeout <0-300 sec>
portmap
add in-port <value> out-port <value>
default-out-port <<value> | disable>
delete <all | rule-id <rule ID>>
override-port <<value> | disable>
tool-bypass <disable | enable>

default-action <decrypt | no-decrypt>

fetch < decryptlist <URL for profile decrypt list file> | nodecryptlist <URL for profile no-decrypt list file>>

fetch replace < decryptlist <URL for profile decrypt list file> | nodecryptlist <URL for profile no-decrypt list file>

ha active-standby <disable | enable>

keymap
add server <server domain name or IP address or IPv6 address> key <key alias>
delete <all | rule-id <rule ID>>

network-group multiple-entry <disable | enable>

no-decrypt tool-bypass <disable | enable>

non-ssl-tcp tool-bypass <disable | enable>

rule add
category <category name> <decrypt | no-decrypt>
domain <domain name string> <decrypt | no-decrypt>
ipv4 <dst | src> <IP address> | ipv6 <dst | src> <IPv6 address> <mask> <decrypt | no-decrypt>
issuer <issuer name string> <decrypt | no-decrypt>

```
      l4port <dst | src> <any | port <value or range>> <decrypt | no-decrypt>
      vlan <any | id <value or range>> <decrypt | no-decrypt>
    rule delete <all | rule-id <rule ID>>
    starttls
      add l4port <port number>
      delete <all | l4port <port number>>
    url-cache miss action <decrypt | defer [timeout <1-10>] | no-decrypt>
  resumption client <disable | enable>
  session debug <disable | enable>
  signing rsa for <primary | secondary> key <key alias>
  trust-store
    fetch <append | replace> <URL for trust store file>
    reset
```

The following table describes the arguments for the **apps inline-ssl** command:

Argument	Description
caching persistence <disable | enable>	Enables or disables caching persistence as follows: • **disable**—Disables caching persistence. • **enable**—Enables caching persistence. The default is **enable**. Disable is recommended only for troubleshooting purposes. For example: **(config) # apps inline-ssl caching persistence disable**
keychain password <password> <confirm password>	Creates an SSL keychain password as follows: **(config) # apps inline-ssl keychain password** Creating a new password for ssl keychain: **Password: ********* **Confirm: ********* The password is used to encrypt all cryptographic materials such as certificates and private keys uploaded to the node. Passwords are not saved on the node. Passwords must be at least 8 characters (up to 64 characters) and must include at least one of each of the following: • uppercase letters • lowercase letters • numbers • special characters **Note**: The keychain password must be configured before installing certificates and keys. If the key has a passphrase, in order to install it, the keychain password and the passphrase must match.
keychain password <password>	Prompts for the SSL keychain password. When keys are installed on the node, you will be prompted to verify the password after any node reboot when you enter configure terminal mode, for example: **# configure terminal** **(config) # apps inline-ssl keychain password required**

Argument	Description
	Enter ssl keychain password: **Password: ***********
keychain password [reset] <password> <confirm password>	Resets an SSL keychain password. When keys are installed on the node, a warning is displayed. **NOTE:** Resetting the password revokes all existing private keys. For example: **(config) # apps inline-ssl keychain password reset** **WARNING: Password is already set. Reset password will revoke all existing private keys.** **Password: ********* Confirm: ***********
version < above | below >	Configures the maximum SSL version and minimum SSL version parameters as follows: • **above**—Configure action for Inline-SSL max-version parameter. • **below**—Configure action for Inline-SSL min-version parameter.
min-version <sslv3 | tls1 | tls11 | tls12 | tls13> max-version <sslv3 | tls1 | tls11 | tls12 | tls13>	Specifies the SSL minimum version and maximum version as follows: • **sslv3**—Specifies SSL 3.0. • **tls1**—Specifies TLS 1.0. • **tls11**—Specifies TLS 1.1. • **tls12**—Specifies TLS 1.2. • **tls13**—Specifies TLS 1.3. The default minimum version is **sslv3**. The default maximum version is **tls12**. Ensure the minimum version is less than the maximum version. For example: **(config) # apps inline-ssl min-version tls11 max-version tls12**
below min-version	Allows or drops below TLS minimum version for the given configuration as follows: • **no-decrypt** - Bypasses below TLS minimum version. • **drop** - Drops below TLS minimum version. The default minimum version is tls1.
above max-version	Allows or drops above TLS maximum version for the given configuration as follows: • **no-decrypt** - Bypasses above TLS maximum version. • **drop** - Drops above TLS maximum version. The default maximum version is tls13.
profile alias <alias> eval-cn <disable | enable>	Specifies an alias to create a policy profile for SSL Decryption for inline tools to specify policy configuration. For example: **(config) # apps inline-ssl profile alias sslprofile**

Argument	Description
	(config apps inline-ssl profile alias sslprofile) #
profile alias <alias> split-proxy [enable | disable]	When you enable the split proxy settings for an inline SSL profile, it divides the TLS connection between the server and client into two independent connections and keeps the security parameters separate. For example: **(config) # apps inline-ssl profile alias sslprofile split-proxy enable**
profile alias <alias> split-proxy server non-pfs-ciphers [enable | disable]	When you enable the non-PFS ciphers settings for an inline SSL profile that has the split proxy settings enabled, it forces the server to use protocols that are lower than TLS1.3 with non-PFS ciphers. This means that the ciphers with DHE/ECDHE key-exchange will be disabled and the server will use only the ciphers with RSA key-exchange. For example: **(config) # apps inline-ssl profile alias sslprofile split-proxy server non-pfs-ciphers enable**
profile alias <alias> tool early-engage [enable | disable]	Allows the inline tools to change the MAC address or VLAN IDs. When a connection request is received from the client, GigaSMART establishes the connection with the inline tool first, before connecting with the server. This helps the inline tools to modify the MAC address or VLAN IDs when sending the traffic back to the server. For additional information and limitations, refer to the *"Tool Early Engage and One-Arm Mode"* section in the *GigaVUE-FM User's Guide*. For example: **(config) # apps inline-ssl profile alias sslprofile tool early-engage enable**
profile alias <alias> tool early-inspect <enable | disable>	Allows the inline tool to view the decrypted data first before connecting to the server. This helps the inline tool to validate the data and ensure that only valid connections are sent to the server. **Note**: You can enable **Tool Early Inspect** only when inbound deployment is supported. For additional information and limitations, refer to the *"Tool Early Inspect"* section in the *GigaVUE-FM User's Guide*. For example: **(config) # apps inline-ssl profile alias ssl_profile inbound tool-early-inspect enable** **(config) # apps inline-ssl profile alias ssl_profile inbound tool-early-inspect connection-timeout 10** **Note**: Connection timeout represents the time by which the tool should respond; if no response is received within the configured interval time, the connections will be reset.
profile alias <alias> one-arm [enable | disable]	Allows both the client and server traffic to travel through the same physical link or logical aggregate port channel.

Argument	Description
	Note: You can enable the **one-arm** mode only if you have enabled the **tool early-engage** option. For additional information and limitations, refer to the *"Tool Early Engage and One-Arm Mode"* section in the *GigaVUE-FM User's Guide*. For example: **(config) # apps inline-ssl profile alias sslprofile one-arm enable**
monitor <disable | enable | inline>	Configures the apps inline-ssl monitoring and SSL decryption/encryption as follows: • **disable**—Disables the monitor mode, and enables SSL decryption/encryption. • **enable**—Enables the monitor mode, and disables SSL decryption/encryption. • **inline**—Enables both the monitor mode, and the SSL decryption/encryption. For example: **(config)# apps inline-ssl profile alias sslprofile monitor disable** **(config)# apps inline-ssl profile alias sslprofile monitor enable** **(config)# apps inline-ssl profile alias sslprofile monitor inline** **Note**: Monitor mode is not supported with clustering.
profile alias <alias> certificate **expired <decrypt | drop>** **invalid <decrypt | drop>** **revocation crl <disable | enable [fail <hard | soft>] [defer timeout <20-100>]>** **revocation ocsp <disable | enable [fail <hard | soft>] [defer timeout <20-100>]>** **self-signed <decrypt | drop>** **unknown-ca <decrypt | drop>**	Configures the handling of expired, invalid, self-signed, and unknown CA certificates as well as enabling or disabling certificate revocation for the profile as follows: • **expired**—Specifies **decrypt** or **drop** for expired certificates. The default is **drop**. ▪ **decrypt**—Accepts the certificate and continues to decryption. ▪ **drop**—Rejects the certificate and drops the connection. • **invalid**—Specifies **decrypt** or **drop** for invalid certificates. The default is **drop**. • **self-signed**—Specifies whether or not to accept self-signed certificates. The default is **drop**. When set to **decrypt**, a new self-signed certificate is generated that matches the identity of the original certificate, but with a different key pair. • **unknown-ca**—Specifies **decrypt** or **drop** for unknown certificate authorities (CA). The default is **drop**. • **revocation**—Enables or disables certificate revocation check as follows: ▪ **crl**—Uses a Certificate Revocation List (CRL) to obtain a list of certificates that have been revoked. ▪ **ocsp**—Uses an Online Certificate Status Protocol to obtain certificate revocation status. ▪ **fail**—Specifies the action to take when the GigaVUE-OS node is unable to perform revocation check or does not already know the revocation status. The options are soft fail and hard fail. With soft fail, the decryption continues, whereas with hard fail, traffic will not be decrypted unless the revocation status is determined for certain. ▪ **defer timeout**—Specifies a deferred action in the profile for the certificate. If the action is **defer**, specify an optional timeout value from 1 to 10 seconds. The default is 1 seconds. GigaSMART will defer the connection until the specified timeout.

Argument	Description
	The revocation check is disabled by default. The connection is permitted, at least until the revocation check returns the status. Examples: **(config apps inline-ssl profile alias sslprofile) # certificate expired decrypt** **(config apps inline-ssl profile alias sslprofile) # certificate invalid drop** **(config apps inline-ssl profile alias sslprofile) # certificate revocation crl disable** **(config apps inline-ssl profile alias sslprofile) # certificate revocation ocsp enable fail soft**
profile alias <alias> client-auth trust-store add <trust-store-alias>	Associates a client trust store with the Inline SSL profile for validating client certificates in inbound and hybrid mTLS deployments. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth trust-store add client_trust_1**
profile alias <alias> client-auth trust-store delete	Removes the client trust store association from the Inline SSL profile so that profile no longer uses that store for client-certificate validation. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth trust-store delete**
profile alias <alias> client-auth primary-signing <key_alias>	Configures the primary signing key pair used to forge client certificates towards the server when client authentication succeeds; required when mTLS is enabled. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth primary-signing client_primary_key**
profile alias <alias> client-auth primary-signing delete	Deletes the primary signing key configuration from the profile; mTLS client authentication cannot be used until a new primary signing key is configured. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth primary-signing delete**
profile alias <alias> client-auth secondary-signing <key_alias>	Configures an optional secondary signing key pair used to forge client certificates when the original client certificate is not trusted by the client trust store (for example, untrusted public CAs). For example: **(config apps inline-ssl profile alias sslprofile) # client-auth secondary-signing client_secondary_key**
profile alias <alias> client-auth secondary-signing	Deletes the secondary signing key configuration from the profile so only the primary signing key is used For example:

Argument	Description
delete	**(config apps inline-ssl profile alias sslprofile) # client-auth secondary-signing delete**
profile alias <alias> client-auth revocation [crl | ocsp] [enable | disable] [fail <hard | soft>] [defer timeout <20–100>]	Configures revocation checking (CRL or OCSP) for client certificates and specifies whether failures are treated as hard or soft errors, with an optional defer timeout in seconds. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth revocation ocsp enable fail soft defer timeout 30**
profile alias <alias> client-auth unknow-ca <decrypt | drop>	Specifies the action when a client certificate is signed by an unknown CA in the client trust store: decrypt continues the connection, while drop terminates it. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth unknow-ca drop**
profile alias <alias> client-auth expired <decrypt | drop>	Specifies the action when the client certificate is expired: decrypt allows the connection to proceed, while drop terminates it. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth expired decrypt**
profile alias <alias> client-auth invalid <decrypt | drop>	Specifies the action when the client certificate is invalid (for example, malformed or missing required attributes): decrypt continues the connection, while drop terminates it. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth invalid drop**
profile alias <alias> client-auth self-signed <decrypt | drop>	Specifies the action when the client certificate is self-signed: decrypt allows the connection, while drop terminates it. For example: **(config apps inline-ssl profile alias sslprofile) # client-auth self-signed drop**
profile alias <alias> clear <decryptlist | nodecryptlist>	Clears the no-decrypt list or the decrypt list for the profile as follows: • **decryptlist**—Clears the decrypt list. • **nodecryptlist**—Clears the no-decrypt list. For example: (config apps inline-ssl profile alias sslprofile) # clear nodecryptlist
profile alias <alias> decrypt tcp inactive-timeout <2-1440 mins>	Specifies additional configuration options for the **decrypt** action for the profile. This is the action to take if the match action is to decrypt as follows: • **tcp**—Specifies the TCP destination for decrypted traffic sent to inline tools. The TCP parameters are as follows:

Argument	Description
timewait-timeout <0-300 sec> **portmap** **add in-port <value> out-port <value>** **default-out-port <<value> | disable>** **delete <all | rule-id <rule ID>** **override-port <<value> | disable>** **tool-bypass <disable | enable>**	○ **inactive-timeout**—Specifies an inactivity timeout from 2 to 1440 minutes. The default is 5 minutes. Proxied connections are terminated when there is no activity for the specified time. ○ **timewait-timeout**—Configure the 'TCP TIMEWAIT' timeout value from the range 0-300 seconds. ○ **portmap**—Specifies the TCP port to use to send to inline tools for a particular destination TCP port from a client as follows: ▪ **add**—Adds a port map by specifying an **in-port** number from 1 to 65535 and an **out-port** number from 1 to 65535. The **in-port** is the TCP destination port from the client. The **out-port** is the port to use to send traffic to the inline tools. There is a maximum of 20 mapping port pairs (**in-port** and **out-port**). ▪ **default-out-port**—Specifies the default out port number from 1 to 65535. This is the TCP port used if the incoming port does not match a configured portmap and if an override port is not configured. ▪ **default-out-port disable**—Disables the default out port configuration. ▪ **delete**—Deletes a specific portmap by its rule ID, or deletes all portmaps. ▪ **override-port <value>**—Specifies the override port number from 1 to 65535. All decrypted traffic to inline tools will use this port as the TCP destination port. ▪ **override-port disable**—Disables the override port configuration. • **tool-bypass**—Specifies whether to bypass the inline tools or not as follows: ▪ **disable**—Specifies not to bypass the inline tools. ▪ **enable**—Specifies to bypass the inline tools. The default is **disable**, which means that all decrypted SSL traffic is sent to the tools. Examples: **(config apps inline-ssl profile alias sslprofile) # decrypt tool-bypass enable** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp inactive-timeout 10** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp portmap override-port disable** **apps inline-ssl profile alias sslprofile decrypt tcp portmap default-out-port 12** Refer to "*Inline SSL Decryption Port Map*" section in the *GigaVUE Fabric Management Guide* for details.
profile alias <alias> default-action <decrypt | no-decrypt>	Specifies the default action for the profile. This is the action to take if none of the rules in the profile match. The actions are as follows: • **decrypt**—Specifies a decrypt action in the profile for default action. • **no-decrypt**—Specifies a no decrypt action in the profile for default action. The default is no-decrypt. Use the default action to create policies such as **decrypt** all but privacy-related

Argument	Description
	categories or **no-decrypt** all but security-related categories. Examples: **(config apps inline-ssl profile alias sslprofile) # default-action decrypt** **(config apps inline-ssl profile alias sslprofile) # default-action no-decrypt**
profile alias <alias> fetch <decryptlist <URL for profile decryptlist file> | nodecryptlist <URL for profile nodecryptlist file>>	Fetches the no-decrypt list or the decrypt list text file for the profile from the specified URL as follows: • **decryptlist <URL>**—Specifies the URL of the decrypt list text file. • **nodecryptlist <URL>**—Specifies the URL of the no-decrypt list text file. No-decrypt list entries are implicitly set to **no-decrypt**, which means that as a policy, no-decrypt listed domains and hostnames will always be bypassed for decryption. As a policy, hostnames or domains matching the decrypt list entries will always be decrypted. No-decrypt list and decrypt list text files must adhere to the following: • In a text (.txt) file, add each domain/FQDN hostname. • Use only the carriage return (newline) to separate entries in a file. Do not use any characters, such as commas or colons, to separate entries in a file. • Each file can contain a maximum of 10,000 entries. Entries beyond 10,000 will be ignored. The supported formats for fetch are: SCP, SFTP, FTP, HTTP. For example: **(config apps inline-ssl profile alias sslprofile) # fetch nodecryptlist http://1.1.1.1/temp/whitelist.txt**
profile alias <alias> fetch replace <decryptlist <URL for profile decryptlist file> | nodecryptlist <URL for profile nodecryptlist file>>	Fetches and replaces the no-decrypt list or the decrypt list text file for the profile from the specified URL as follows: • **decryptlist <URL>**—Specifies the URL of the decrypt list text file. • **nodecryptlist <URL>**—Specifies the URL of the no-decrypt list text file.
profile alias <alias> ha active-standby <disable | enable>	Enables GigaSMART inline network high availability (HA) active standby support. When there is an inline SSL network group topology with two network port pairs (Na1, Nb1 and Na2, Nb2), incoming traffic from one network (for example, Na1) may change to another network (for example, Na2) due to upstream devices, such as firewalls performing high availability active standby failover. The options are as follows: • **disable**—Disables HA active standby support. • **enable**—Enables HA active standby support. When enabled, GigaSMART will forward traffic to the correct inline network if an upstream device fails over. The default is disable.

Argument	Description
	For example: **(config apps inline-ssl profile alias sslprofile) # ha active-standby enable**
profile alias <alias> keymap add server <server domain name or IP address or IPv6 address> key <key alias>	Creates an SSL server key map, which creates a key map entry. A server key map is for an inbound deployment of SSL Decryption for inline tools, in which the customer has the server keys. A server key map binds keys from the keystore as follows: • **server**—Specifies the domain name of the server or the IP address of the server or the IPv6 address of the server. **NOTE:** IPV6 traffic decryption is supported only for GEN 3 cards. Refer to the GigaVUE-HC1 Hardware Installation Guide and GigaVUE-HC3 Hardware Installation Guide for the list of GEN 3 card numbers. • **key**—Specifies the alias of the key in the keystore. The key alias is a key pair generated by the **apps keystore** command. To bind with the key map, a private key and a certificate are required. The maximum number of key mappings is 1000. Examples: **(config apps inline-ssl profile alias sslprofile) # keymap add server server_1 key server_1_key** **(config apps inline-ssl profile alias sslprofile) # keymap add server server_2 key server_2_key** **(config apps inline-ssl profile alias sslprofile) # keymap add server server_3 key server_3_key** **(config apps inline-ssl profile alias sslprofile) # keymap add server server_4 key server_4_key** **Note**: Use the **apps keystore** command to add server keys to the key store. Refer to apps keystore.
profile alias <alias> keymap delete <all | rule-id <rule ID>	Deletes an SSL server key map entry, either all key maps or a specific key map by its ID. Examples: **(config apps inline-ssl profile alias sslprofile) # keymap delete all** **(config apps inline-ssl profile alias sslprofile) # keymap delete rule-id 12** **NOTE:** Manual assignment of key map IDs is not supported. When a key map is deleted, the remaining IDs may become misaligned, potentially causing decryption failures or unintended bypasses. To ensure sequential integrity and reliable decryption, it is recommended to delete all key maps and reconfigure them.
profile alias <alias> network-group	Enables or disables inline network group multiple entry for the profile. The default is disabled.

Argument	Description
multiple-entry <disable | enable>	An inline network group topology can have multiple network port pairs (for example, Na1, Nb1 and Na2, Nb2). With multiple network port pairs, traffic from a network interface might traverse GigaSMART multiple times. Intercepted traffic from GigaSMART might reenter GigaSMART through a different network interface within the same network group. Starting in software version 5.3, the same traffic sent from GigaSMART can reenter GigaSMART. GigaSMART remembers the inline incoming inline network interface (for example, Na1) for each connection. When traffic from the same connections reaches GigaSMART with a different inline network interface (for example, Na2) within the same network group, GigaSMART will forward the traffic to the corresponding opposite network interface (for example, Nb2), without further processing. This allows traffic from the same connection to reenter GigaSMART. However, the same traffic sent by GigaSMART reentering through the same network port pair (for example, Nb2, Na2) is not supported. For example: **(config apps inline-ssl profile alias sslprofile) # network-group multiple-entry enable**
profile alias <alias> no-decrypt tool-bypass <disable | enable>	Specifies additional configuration options for the **no-decrypt** action for the profile. This is the action to take if the match action is to bypass decryption as follows: • **tool-bypass**—Specifies whether to bypass the inline tools or not as follows: ▪ **disable**—Specifies to send traffic to the inline tools (not to bypass it). ▪ **enable**—Specifies to bypass the inline tools. For example, traffic that is not decrypted does not need to go to the inline tools. The default is **disable**, which means that all non-decrypted SSL traffic is sent to the tools. For example: **(config apps inline-ssl profile alias sslprofile) # no-decrypt tool-bypass enable**
profile alias <alias> non-ssl-tcp tool-bypass <disable | enable>	Specifies a non-SSL TCP action as follows: • **tool-bypass**—Specifies whether to bypass the inline tools or not as follows: ▪ **disable**—Disables SSL profile configuration on non-SSL TCP bypass to network port. ▪ **enable**—Enables SSL profile configuration on non-SSL TCP bypass to network port. The default is **disable**, which means that all non-SSL traffic is sent to the tools. For example: **(config apps inline-ssl profile alias sslprofile) # non-ssl-tcp tool-bypass enable**
profile alias <alias> rule add	Configures rules for the profile based on attributes to match. Select decrypt or no decrypt. The maximum number of rules that can be added is 128, regardless of type. The

Argument	Description
category <category name> <decrypt | no-decrypt> domain <domain name string> <decrypt | no-decrypt> ipv4 <src | dst> <IP address> ipv6 <dst | src> <IPv6 address> <mask> <decrypt | no-decrypt> issuer <issuer name string> <decrypt | no-decrypt> l4port <src | dst> <any | port <value or range>> <decrypt | no-decrypt> vlan <any | id <value or range>> <decrypt | no-decrypt>	rule types are as follows: o **category**—Specifies a rule based on the URL category of the destination hostname in the Server Name Indication (SNI) extension. There are dozens of category names from which to choose. For example, you can create a no decrypt policy for privacy-related categories, such as health care, financial, education, and government. The categories are resolved by a third party database, Webroot. o **domain**—Specifies a rule based on the destination hostname or domain name from the SNI. o **ipv4**—Specifies a rule based on IPv4 address and netmask for either source or destination. o **ipv6**—Specifies a rule based on IPv6 address and netmask for either source or destination. **NOTE:** IPV6 traffic decryption is supported only for GEN 3 cards. Refer to the GigaVUE-HC1 Hardware Installation Guide and GigaVUE-HC3 Hardware Installation Guide for the list of GEN 3 card numbers. o **issuer**—Specifies a rule based on issuer of the server X.509 certificate. For example, an issuer name has the following format: /C=US/ST=ca/L=santa clara/O=gigamon/OU=eng/CN=RootCA/emailAddress=john.doe@gigamon o **l4port**—Specifies a rule based on any Layer 4 (L4) port for either source or destination, for a specific L4 port number or range from 0 to 65535. o **vlan**—Specifies a rule based on any VLAN ID or a specific VLAN ID or range from 0 to 4095. Examples: **(config apps inline-ssl profile alias sslprofile) # rule add domain domain1.com no-decrypt** **(config apps inline-ssl profile alias sslprofile) # rule add category search_engines decrypt** **(config apps inline-ssl profile alias sslprofile) # rule add ipv4 src 1.1.1.1 mask 255.255.0.0 no-decrypt** **(config apps inline-ssl profile alias sslprofile) # rule add l4port src port 443 decrypt** **(config apps inline-ssl profile alias sslprofile) # rule add vlan id 100.200 no-decrypt** **(config apps inline-ssl profile alias sslprofile) # rule add ipv6 dst 3000::1 mask FFFF::0 decrypt**
profile alias <alias> rule delete <all | rule-id <rule ID>	Deletes rules for the profile, either all rules or a specific rule by its rule ID. Examples: **(config apps inline-ssl profile alias sslprofile) # rule delete all** **(config apps inline-ssl profile alias sslprofile) # rule delete rule-id 2**
profile alias <alias>	Specifies StartTLS Layer 4 (L4) ports as follows:

Argument	Description
starttls **add l4port <port number>** **delete <all | l4port <port number>**	• **add**—Specifies an L4 port number to add. • **delete**—Specifies an L4 port number to delete or all L4 ports. The specific ports to monitor StartTLS traffic must be specified for the profile. Up to 20 ports can be specified in a comma separated list. **NOTE:** Both HTTP CONNECT and StartTLS are supported using the same **starttls** command. In HTTP CONNECT, the L4 port is the explicit proxy port number. Examples: **(config apps inline-ssl profile alias sslprofile) # starttls add l4port 44** **(config apps inline-ssl profile alias sslprofile) # starttls delete all** **(config apps inline-ssl profile alias sslprofile) # starttls delete l4port 12**
profile alias <alias> url-cache miss action <decrypt | defer [timeout <1-10>] | no-decrypt>	Specifies an action to take for the profile. This is the action to take on the traffic if GigaSMART is unable to resolve the URL category information locally. The actions are as follows: • **decrypt**—Specifies a decrypt action in the profile for URL cache. • **defer [timeout]**—Specifies a deferred action in the profile for URL cache miss. If the action is **defer**, specify an optional timeout value from 1 to 10 seconds. The default is 1 second. GigaSMART will defer the connection until the specified timeout before reevaluating the policy. • **no-decrypt**—Specifies a no decrypt (bypass) action in the profile for URL cache miss. The default is **no-decrypt**. Examples: **(config apps inline-ssl profile alias sslprofile) # url-cache miss action decrypt** **(config apps inline-ssl profile alias sslprofile) # url-cache miss action defer** **(config apps inline-ssl profile alias sslprofile) # url-cache miss action defer timeout 5**
resumption client <enable | disable>	Enables or disables client initiated resumption as follows: • **disable**—Disables resumption on client. • **enable**—Enables resumption on client. The default is **enable**. For example: **(config) # apps inline-ssl resumption client disable**

Argument	Description
session debug [disable | enable]	Reserved for internal use.
signing rsa for <primary | secondary> key <key alias>	Specifies SSL signing for RSA. For SSL certificate re-signing, there are different CAs used (primary and secondary) as follows: • **primary**—(Mandatory) Specifies the primary signing certificate for RSA. The primary CA re-signs certificates for servers that present a valid certificate. • **secondary**—(Optional) Specifies the secondary signing certificate for RSA. The secondary CA re-signs certificates for servers that are invalid or that fail validation. **Note**: If **decrypt** is specified for invalid certificates, the primary certificate will be used for re-signing invalid certificates if the secondary certificate has not been configured. • **key**—Specifies the alias of the key in the keystore. The key alias from the keystore can be a Man-in-the-Middle (MitM) key pair or a self-signed generated certificate. Refer to apps keystore. **NOTES:** • For SSL certificate re-signing, the subject name is copied from the original certificate. • The validation period for re-signed certificates is one week. Examples: **(config) # apps inline-ssl signing rsa for primary key issl1-primary-ca** **(config) # apps inline-ssl signing rsa for secondary key issl1-secondary-ca**
trust-store <fetch <append | replace> <URL for trust store file> | reset>	Installs trusted certificate authority (CA) for server certificate validation as follows: • **fetch append**—Fetches the CA at the specified URL for the inline SSL trust store file and appends it to the end of the existing trust store. The URL must point to a file stored in PEM format. • **fetch replace**—Fetches the CA at the specified URL for the inline SSL trust store file and replaces the existing trust store. The URL must point to a file stored in PEM format. • **reset**—Resets to the default trusted CAs for server certification validation. The supported formats for fetch are: SCP, SFTP, FTP, HTTP. **Note**: Revoked Certificates can be removed using **no apps inline-ssl trust-store certificate fingerprint** <> CLI command. Examples: **(config) # apps inline-ssl trust-store fetch replace http://1.1.1.1/mitm/my_trust_store.pem** **(config) # apps inline-ssl trust-store fetch append http://1.1.1.1/mitm/my_trust_store.pem** **(config) # apps inline-ssl trust-store reset**

Related Commands

The following table summarizes other commands related to the **apps inline-ssl** command:

Task	Command
Displays inline SSL persistent cache entries that match the certificate common name (CN).	**# show apps inline-ssl caching certificate validation internal_ca1.com**
Displays inline SSL persistent certificate cache status, including the number of entries saved in the database.	**# show apps inline-ssl caching certificate validation status**
Displays inline SSL persistent cache entries that match URL domain name.	**# show apps inline-ssl caching url www.gigamon.com**
Displays inline SSL persistent URL cache status, including the number of records cached and the database version.	**# show apps inline-ssl caching url status**
Displays all inline SSL global parameters.	**# show apps inline-ssl global**
Displays brief information for 1000 inline SSL monitor mode sessions.	**# show apps inline-ssl monitor session any**
Displays brief information for inline SSL monitor mode sessions, based on the match.	**# show apps inline-ssl monitor session match ipv4-src 192.168.43.75/32 ipv4-dst 126.1.0.101/32 l4port-src 1124 l4port-dst 443**
Displays inline SSL monitor mode session summary.	**# show apps inline-ssl monitor summary**
Displays a specified inline SSL profile including mTLS client-auth settings such as Client Certificate Trust-store and client-auth primary/secondary signing keys.	**# show apps inline-ssl profile alias sslprofile**
Displays domain name entry if it is in the decrypt list.	**# show apps inline-ssl profile alias sslprofile decryptlist BadCo.com**
Displays domain name entry if it is in the no-decrypt list.	**# show apps inline-ssl profile alias sslprofile nodecryptlist GoodCo.com**
Displays all inline SSL profiles.	**# show apps inline-ssl profile all**
Displays any inline SSL session including client-authentication state for mTLS (for example, Client:OK, Client:REJECTED, or Client:NOT_DONE).	**# show apps inline-ssl session any**
Reserved for internal use.	**# show apps inline-ssl session debug**
Displays inline SSL sessions that match any IPv4 source IP address and mask, any IPv4 destination IP address and mask, any L4 source and destination port, and hostname.	**# show apps inline-ssl session match ipv4-src any ipv4-dst any l4port-src any l4port-dst any hostname gigamon.com**
Displays inline SSL sessions that match a specific IPv4 source IP address and mask, a specific IPv4 destination IP address and mask, any L4 source and destination port, and hostname	**# show apps inline-ssl session match ipv4-src 126.1.0.141/21 ipv4-dst 126.1.0.22/29 l4port-src any l4port-dst any hostname gigamon.com**
Displays inline SSL sessions that match a specific IPv4 source IP address and mask, destination IP address and mask, L4 source port number and L4 destination port number, and hostname.	**# show apps inline-ssl session match ipv4-src 192.168.1.1/24 ipv4-dst 192.168.1.2/24 l4port-src 56708 l4port-dst 443 hostname gigamon.com**
Displays inline SSL sessions that match a specific IPv4 source IP address and mask, destination IP address and	**# show apps inline-ssl session match ipv4-src 192.168.1.1/24 ipv4-dst 192.168.1.2/24 l4port-src**

Task	Command
mask, L4 source port number and L4 destination port number, and hostname in detail.	**56708 l4port-dst 443 hostname gigamon.com detail**
Displays inline SSL sessions that match a hostname. Not all the matching criteria needs to be specified, for example, instead of gigamon.com, you can specify gigamon or gamon.	**# show apps inline-ssl session match hostname gigamon.com# show apps inline-ssl session match hostname gigamon# show apps inline-ssl session match hostname gamon**
Displays inline SSL sessions that match a hostname in detail.	**# show apps inline-ssl session match hostname gigamon.com detail**
Displays inline SSL session summary information ,including aggregated client certificate counters for mTLS (Expired, Self-Signed, Unknown CA, Certificate Accepted, Certificate Rejected, and related revocation stats).	**# show apps inline-ssl session summary**
Displays inline SSL trust store.	**# show apps inline-ssl trust-store all**
Displays a specified inline SSL certificate by fingerprint. The format is XX:XX:XX:XX, which is the hex representation of the first four octets of the certificate's SHA1 fingerprint.	**# show apps inline-ssl trust-store certificate fingerprint D1:EB:23:A4**
Deletes a specified inline SSL profile.	**(config) # no apps inline-ssl profile alias sslprofile**
Deletes all inline SSL profiles.	**(config) # no apps inline-ssl profile all**
Specifies that SSL primary and secondary certificate for RSA can be overwritten. **Note**: The primary and secondary signing keys are not deleted with these commands, however, after these commands are issued, a new certificate/key pair can be configured, which will overwrite the existing certificate/key pair.	**(config) # no apps inline-ssl signing rsa for primary** **(config) # no apps inline-ssl signing rsa for secondary**
Deletes a specified inline SSL certificate by fingerprint.	**(config) # no apps inline-ssl trust-store certificate fingerprint 8E:1C:74:F8**
Clears the inline SSL certificate validation persistent cache.	**(config) # clear apps inline-ssl caching cert-validation**
Clears the inline SSL URL persistent cache.	**(config) # clear apps inline-ssl caching url**
Clears inline SSL session statistics summary.	**(config) # clear apps inline-ssl session summary**

apps keystore

Use the **apps keystore** command to download and assign RSA keys and key pairs. If certificates are in the keystore, no re-signing is needed. The keystore can contain a maximum of 4000 keys.

Inline SSL decryption requires a key pair, which includes both private and public keys (leaf certificate and CA certificate chain).

Passive SSL decryption and Hardware Security Module (HSM) require only the private key.

The **apps keystore** command has the following syntax:

```
apps keystore
  rsa | ecdsa|pq <key alias>
  certificate <download url <download URL> | key-str <key string>>
  comment <comment>
  pkcs12 <download url <download URL> [password <password>]>
  private-key <download url <download URL> | key-str <key string>> [password <PEM
password> | type hsm]
  self-signed
    common-name <CN>
    country <C>
    hash-type <SHA-1 | SHA-256 | SHA-384 | SHA-512>
    keysize <1024 | 2048 | 4096>
    org-name <O>
    org-unit <OU>
    state <S>
    valid <number of days>
```

The following table describes the arguments for the **apps keystore** command:

Argument	Description
rsa | ecdsa | pq<key alias>	Specifies the following key alias: • RSA • ECDSA • PQ
certificate <download url <download URL> | key-str <key string>>	Downloads a certificate or cuts and pastes a certificate. Use this command to configure the Man-in-the-Middle (MitM) primary CA or optional secondary CA as follows: • **url**—Specifies the download URL for the certificate PEM file. • **key-str**—Specifies the SSL key PEM file by providing a key string for a certificate. Enclose the key string in double quotation marks. The download URL specifies an SSL certificate. The supported formats for download are HTTP, FTP, SCP, and SFTP. For example, to download a certificate: **(config) # apps keystore rsa issl1-primary-ca certificate download url http://1.1.1.2/mitm/primary_ca.cert** **(config) # apps keystore rsa issl1-secondary-ca certificate download url http://1.1.1.2/mitm/secondary_ca.cert** For example, to cut and paste a certificate, specify the private key string in PEM format: **(config) # apps keystore rsa key1 certificate key-str "-----**

Argument	Description
	BEGIN RSA PRIVATE KEY----- ...-----END RSA PRIVATE KEY----- " To bind the certificate to the primary CA: **(config) # apps inline-ssl signing for primary key <key alias>** Refer to apps inline-ssl.
comment <comment>	Adds a comment to an RSA keystore key pair. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # apps keystore rsa key1 comment "This is a comment"**
pkcs12 <download url <download URL> [password <password>]>	Downloads a PKCS12 file containing the private key and the certificate as follows: • **url**—Specifies the download URL for PKCS12, pfx file. • **password**—Specifies an optional password for PKCS12. If a password is not specified after the **password** keyword, you will be prompted for it. The download URL specifies a PKCS12 container. The supported formats for download are HTTP, FTP, SCP, and SFTP. For example: **(config) # apps keystore rsa key2 pkcs12 download** url sftp://test:mytest@10.10.10.10/home/test/ssldecrypt/keys/srv1k.pfx **(config) # apps keystore ecdsa key2 pkcs12 download** url sftp://test:mytest@10.10.10.10/home/test/ssldecrypt/keys/srv1k.pfx Refer to Supported Algorithms for details on the compatible algorithms to download a PKCS12 file.
private-key <download url <download URL> | key-str <key string>> [password <PEM password> | type hsm]	Downloads a private key or cuts and pastes a private key. Use this command to configure the MitM primary CA or optional secondary CA as follows: • **url**—Specifies the download URL for the private key PEM file. • **key-str**—Specifies the SSL key PEM file by providing a key string for a private key. Enclose the key string in double quotation marks. • **password**—Specifies a password for the PEM private key if it is encrypted. Otherwise, the PEM private key needs to be decrypted. • **type**—Specifies the key type for keys residing on HSM. The only value is **hsm**. The download URL specifies an SSL private key. The supported formats for download are HTTP, FTP, TFTP, SCP, and SFTP. For example, to download a private key: **(config) # apps keystore rsa issl1-primary-ca private-key** download url http://1.1.1.1/mitm/primary_ca.key **(config) # apps keystore rsa issl1-secondary-ca private-key download url http://1.1.1.2/mitm/secondary_ca.key**

Argument	Description
	(config) # apps keystore ecdsa issl1-primary-ca private-key download url http://1.1.1.1/mitm/primary_ca.key **(config) # apps keystore ecdsa issl1-secondary-ca private-key download url http://1.1.1.2/mitm/secondary_ca.key** For example, to cut and paste a private key, specify the key string in PEM format: **(config) # apps keystore rsa key1 private-key key-str "----- text-----"** **(config) # apps keystore ecdsa key1 private-key key-str "----- text-----"** To bind the private key to the primary CA: **(config) # apps inline-ssl signing for primary key <key alias>** For example, to download an encrypted private key when the password is specified on the command line: **(config) # apps keystore rsa K4 private-key** download url http://dominos.gigamon.com/~ama/misc/encrypted_pkey.pem password admin1100.0% [## For example, to download an encrypted private key when the password is not specified on the command line, you will be prompted for the passphrase as follows: **(config) # apps keystore rsa K4 private-key** download url http://dominos.gigamon.com/~ama/misc/encrypted_pkey.pem100.0% [##PEM Passphrase: ********** Refer to apps inline-ssl. For example, to configure keys residing on HSM: **(config) # apps keystore rsa mykey private-key download url http://10.115.0.100/tftpboot/myname/hsm/key_pkcs11_ua88af6e573c9c6c39b245a15edfc3ebcbebbdae4f type hsm** Refer to apps hsm.
self-signed **common-name <CN>** **country <C>** **hash-type <SHA-1 | SHA-256 | SHA-384 | SHA-512>** **keysize <1024 | 2048 | 4096>** **org-name <O>** **org-unit <OU>** **state <S>** **valid <number of days>**	Generates a self-signed certificate and key (key pair) as follows: • **common-name <CN>**—Specifies the common name for the certificate. • **country <C>**—Specifies the country name for the certificate. • **hash-type**—Specifies the type of hashing for the certificate. The values are: SHA-1, SHA-256, SHA-384, and SHA-512. • **key-size**—Specifies the key size for the certificate. The values are 1024, 2048, and 4096. • **org-name <O>**—Specifies the organization name for the certificate. • **org-unit <OU>**—Specifies the organizational unit name for the certificate. • **state <S>**—Specifies the state for the certificate. • **valid**—Specifies the number of days for which the certificate is valid.

Argument	Description
	The range is from 1 to 2000 days. The **common-name** and **org-name** are mandatory. The generated key and certificate will be stored as an entry in the keystore. The key can be imported into a primary or secondary signing key for SSL Decryption for inline tools. For example: **(config) # apps keystore rsa internal-ca1 self-signed common-name internal_ca1.com country US state CA org-name GIMO org-unit ENG keysize 2048 hash-type SHA-256 valid 100** **(config) # apps keystore ecdsa internal-ca1 self-signed common-name internal_ca1.com country US state CA org-name GIMO org-unit ENG keysize 2048 hash-type SHA-256 valid 100** To bind the key to use with the primary or secondary signing key: **(config) # apps inline-ssl signing rsa for primary key <key alias>** Refer to apps inline-ssl.

NOTE: If you select your key as type **'PQ'**, you will not be able to generate a self signed PQC certificate.

Related Commands

The following table summarizes other commands related to the **apps keystore** command:

Task	Command
Displays a certificate for a specified SSL key.	**# show apps keystore alias primary certificate**
Displays certificate with PQ related details.	**#show apps keystore alias <key-alias> certificate**
Displays a summary for a specified SSL key.	**# show apps keystore alias primary summary**
Displays all SSL keys.	**# show apps keystore all**
Deletes specified ecdsa keys from the keystore	**(config) # no apps keystore ecdsa aliasprimary**
Deletes all ecdsa keys from the keystore	**(config) # no apps keystore ecdsa all**
Deletes a specified SSL key.	**(config) # no apps keystore rsa aliasprimary**
Deletes all SSL keys.	**(config) # no apps keystore rsa all**

Supported Algorithms

The following algorithms are supported when downloading a PKCS12 file containing the private key and the certificate:

- PBE-SHA1-RC4-128
- PBE-SHA1-RC4-40
- PBE-SHA1-3DES
- PBE-SHA1-2DES
- PBE-SHA1-RC2-128
- PBE-SHA1-RC2-40

apps listener

Use the **apps listener** command to configure the listener.

The **apps listener** command has the following syntax:

```
apps listener alias <alias>
 type <gtp-cups | tunnel>
 l4
 port-list
 protocol <tcp | udp>
l3
 protocol <ipv4 | ipv6 | dual>
 ttl <1-255>  dscp <0-63>
mode l3 <promiscuous | interface>
gsgroup <add | delete>
```

The following table describes the arguments for the **apps listener** command:

Argument	Description
alias <alias>	Specifies the listener alias.
type <gtp-cups | tunnel>	Specifies the type of the listener as follows: • **gtp-cups** — GigaSMART host for GTP-Cups. • **tunnel**— GigaSMART host for TCP Tunnel.
l4 << port-list> **<protocol <tcp | udp>>**	Configures the following transportation layer parameters: • **port-list**— Configure listener source port list. • **protocol** — Configures the following listener l4 protocol o TCP o UDP

Argument	Description
l3 <protocol <ipv4 | ipv6 | dual> **ttl <1-255> dscp <0-63>**>	Configures the following network layer parameters: • **dscp** — Configures the DSCP value. The value ranges from 0 to 63. • **protocol** — Configures the following listener protocol. o IPv4 — Uses IPv4 for communication. o IPv6 — Uses IPv6 for communication. o Dual — Provides Dual stack support. • **ttl** — Configures the Time-To-Live (TTL) in seconds. The value ranges from t to 255.
mode <l3 <promiscuous | interface>>	Configures the listener mode as follows: o **promiscuous** — Configures to promiscuous mode. o **interface** — Configures to listen to interface mode. **Note**: Promiscuous node is not supported.
gsgroup <add | delete>	Configures the gsgroup on which the listener has to be activated or deactivated as follows: • **add** — Activates the listener on which the gsgroup is configured. • **delete** —Deactivates the listener on which the gsgroup is configured.

apps netflow

The following table describes the arguments for the **apps netflow** command:

Argument	Description
exporter	Configures the NetFlow Exporter.
monitor	Configures the NetFlow Monitor.
record	Configures the NetFlow Record.

The following table describes the syntax details of the command:

Command	Argument	Description
apps netflow		Configures the NetFlow Generation parameters.
	exporter	Configures the NetFlow Generation Exporter.
	monitor	Configures the NetFlow Generation Monitor.
	record	Configures the NetFlow Generation Record.
apps netflow exporter		Specifies the NetFlow Generation Exporter.
	alias	Configures an alias for the NetFlow Generation Exporter.

Command	Argument	Description
apps netflow exporter alias		Specifies a NetFlow Exporter alias.
	<alias>	Specifies the name for the NetFlow Generation Exporter.
apps netflow exporter alias <alias>		Specifies the NetFlow Generation Exporter alias parameters.
	<cr>	Enters the *NetFlow Generation Exporter Mode* to configure a NetFlow Generation Exporter.
	description	Specifies a description for the NetFlow Generation Exporter (optional).
	destination ipv4	Specifies a destination IPv4 address.
	destination ipv6	Specifies a destination IPv6 address. In an exporter, you can either configure IPv4 or IPv6 address.
	dscp	Configures DSCP parameters for datagrams sent by the NetFlow Generation Exporter (optional).
	filter	Specifies NetFlow Exporter filters.
	format	Configures NetFlow Generation Exporter formats.
	snmp enable	Enables SNMP for a specified NetFlow exporter. When enabled, SNMP requests that are sent by the external exporter will be replied so that external NetFlow collectors can integrate with GigaSMART NetFlow Generation. The default port is the SNMP UDP port. The default port number is 161.
	template-refresh-interval	Specifies the NetFlow Generation template and option template refresh interval.
	transport	Specifies the NetFlow Generation Transport Protocol.
	ttl	Configures the NetFlow Generation Time-To-Live (TTL) in seconds (optional).
apps netflow exporter alias <alias> description		Specifies a description for the NetFlow Generation Exporter.
	<string>	Specifies the description of the NetFlow Generation Exporter.
apps netflow exporter alias <alias>		Specifies an alias destination for the NetFlow Generation Exporter.

Command	Argument	Description
destination		
	ip4addr	Configures the IPv4 address of the NetFlow Generation Collector.
apps netflow exporter alias <alias> destination ip4addr		Specifies the NetFlow Generation Exporter destination IP address for the NetFlow Generation Collector.
	<IPv4 address>	Specifies the NetFlow Generation IPv4 address.
apps netflow exporter alias <alias> dscp		Specifies the NetFlow Generation DSCP parameters.
	<0-63>	Specifies NetFlow Generation DSCP parameters. The default is 0.
apps netflow exporter alias <alias> filter		Specifies NetFlow exporter filtering rules.
	add pass	Adds a new exporter pass filter.
	delete	Deletes an existing exporter filter.
apps netflow exporter alias <alias> filter add pass input interface range apps netflow exporter alias <alias> filter add pass input interface value	<bid/sid/pid_x..pid_y> <bid/sid/pid>	Adds a new exporter pass filter rule for an input interface as follows: • **range**—Configures input interface port range as <bid/sid/pid_x..pid_y>. • **value**—Configures input interface port value as <bid/sid/pid>.
apps netflow exporter alias <alias> filter add pass ipv4 dcsp	any value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef> | pos <1-3>	Adds a new exporter pass filter rule for IPv4 DiffServ Code Point (DSCP) bits as follows: • **any value**—Configures IPv4 DSCP bits value for Assured Forwarding Class 1 to Class 4 with Low, Med, or High Drop, or for Expedited Forwarding. • **pos**—Configures IPv4 DSCP bits position.
apps netflow exporter alias <alias> filter add pass ipv4 dst	any [range <ipv4_ address..ipv4_address>] | value <IP address>] | pos <1-3>	Adds a new exporter pass filter rule for IPv4 destination address as follows: • **any range**—Configures IPv4 destination address range as <ipv4_address..ipv4_address>.

Command	Argument	Description
		• **any value**—Configures IPv4 destination address value as <IP address>. • **pos**—Configures IPv4 destination address position.
apps netflow exporter alias <alias> filter add pass ipv4 protocol	any [range <1-byte-hex..1-byte-hex> | value [icmp-ipv4 | igmp | ipv4ov4 | tcp | udp | ipv6 | rsvp | gre | <1-byte-hex>] | pos <1-3>	Adds a new exporter pass filter rule for IPv4 protocol as follows: • **any range**—Configures IPv4 protocol range as <1-byte-hex..1-byte-hex>. • **any value**—Configures IPv4 protocol value as a name or as <1-byte-hex>. • **pos**—Configures IPv4 protocol position.
apps netflow exporter alias <alias> filter add pass ipv4 src	any [range <ipv4_address..ipv4_address>] | [value <IP address>] | pos <1-3>	Adds a new exporter pass filter rule for IPv4 source address as follows: • **any range**—Configures IPv4 source address range as <ipv4_address..ipv4_address>. • **any value**—Configures IPv4 source address value as <IP address>. • **pos**—Configures IPv4 source address position.
apps netflow exporter alias <alias> filter add pass ipv4 tosval	any [range <1-byte-hex..1-byte-hex> | value <1-byte-hex>] | pos <1-3>	Adds a new exporter pass filter rule for IPv4 Type of Service (TOS) as follows: • **any range**—Configures IPv4 type of service range as <1-byte-hex..1-byte-hex>. • **any value**—Configures IPv4 type of service value as <1-byte-hex>. • **pos**—Configures IPv4 type of service position.
apps netflow exporter alias <alias> filter add pass ipv6 dcsp	any value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef> | pos <1-3>	Adds a new exporter pass filter rule for IPv6 DiffServ Code Point (DSCP) bits as follows: • **any value**—Configures IPv6 DSCP bits value for Assured Forwarding Class 1 to Class 4 with Low, Med, or High Drop, or for Expedited Forwarding. • **pos**—Configures IPv6 DSCP bits position.
apps netflow exporter alias <alias> filter add pass ipv6 dst	any [range <ipv6_address..ipv6_address>] | [value <ipv6 address>] | pos <1-3>	Adds a new exporter pass filter rule for IPv6 destination address as follows: • **any range**—Configures IPv6 destination address range as <ipv6_address..ipv6_address>. • **any value**—Configures IPv6 destination address value as <ipv6 address>. • **pos**—Configures IPv6 destination address position.
apps netflow exporter alias <alias> filter add pass ipv6 flow-label	any [range <3-byte-hex..3-byte-hex> | [value <3-byte-hex>] | pos <1-3>	Adds a new exporter pass filter rule for IPv6 flow label as follows: • **any range**—Configures IPv6 flow label range as <3-byte-hex..3-byte-hex>.

Command	Argument	Description
		• **any value**—Configures IPv6 flow label value as <3-byte-hex>. • **pos**—Configures IPv6 flow label position.
apps netflow exporter alias <alias> filter add pass ipv6 src	any [range <ipv6_ address..ipv6_address>] | [value <ipv6 address>] | pos <1-3>	Adds a new exporter pass filter rule for IPv6 source address as follows: • **any range**—Configures IPv6 source address range as <ipv6_ address..ipv6_address>. • **any value**—Configures IPv6 source address value as <ipv6 address>. • **pos**—Configures IPv6 source address position.
apps netflow exporter alias <alias> filter add pass l4port dst	any [range <x..y>] | [value <0-65535>] | pos <1-3>	Adds a new exporter pass filter rule for Layer 4 (L4) destination port as follows: • **any range**—Configures L4 destination port range as a range between 0 and 65535, <x..y>. • **any value**—Configures L4 destination port value as a number between 0 and 65535. • **pos**—Configures L4 destination port position.
apps netflow exporter alias <alias> filter add pass l4port src	any [range <x..y>] | [value <0-65535>] | pos <1-3>	Adds a new exporter pass filter rule for Layer 4 (L4) source port as follows: • **any range**—Configures L4 source port range as a range between 0 and 65535, <x..y>. • **any value**—Configures L4 source port value as a number between 0 and 65535. • **pos**—Configures L4 source port position.
apps netflow exporter alias <alias> filter add pass mac dst	any [range <MAC_ address..MAC_address>] | [value <MAC_address>] | pos <1-3>	Adds a new exporter pass filter rule for MAC destination address as follows: • **any range**—Configures MAC address range as <MAC_ address..MAC_address>. • **any value**—Configures MAC address value as <MAC_ address>. • **pos**—Configures MAC address position.
apps netflow exporter alias <alias> filter add pass mac src	any [range <MAC_ address..MAC_address>] | [value <MAC_address>] | pos <1-3>	Adds a new exporter pass filter rule for MAC source address as follows: • **any range**—Configures MAC address range as <MAC_ address..MAC_address>. • **any value**—Configures MAC address value as <MAC_ address>. • **pos**—Configures MAC address position.
apps netflow exporter alias <alias> filter add pass vlan id	any [range <vlan1..vlan2>] | [value <1-4094>] | pos <1-4>	Adds a new exporter pass filter rule for VLAN ID as follows: • **any range**—Configures VLAN ID range as <vlan1..vlan2>. • **any value**—Configures VLAN ID value as a number between

Command	Argument	Description
		1 and 4094. • **pos**—Configures VLAN ID position.
apps netflow exporter alias <alias> filter delete	all | filter-id <ID>	Deletes all existing filters on this exporter or deletes a filter by filter ID.
apps netflow exporter alias <alias> format		Specifies NetFlow Generation Exporter formats.
apps netflow exporter alias <alias> format cef version		Specifies NetFlow Generation Exporter Common Event Format (CEF).
	23	Specifies CEF version 23.
apps netflow exporter alias <alias> format netflow version		Specifies NetFlow Generation Exporter versions.
	ipfix	Specifies IPFIX.
	netflow-v5	Specifies version 5 (v5).
	netflow-v9	Specifies version 9 (v9). This is the default NetFlow version.
apps netflow exporter alias <alias> snmp	enable	Enables SNMP for a specified NetFlow exporter. When enabled, SNMP requests that are sent by the external exporter will be replied so that external NetFlow collectors can integrate with GigaSMART NetFlow Generation. The default port is the SNMP UDP port. The default port number is 161.
apps netflow exporter alias <alias> template-refresh-interval		Specifies the NetFlow Generation template and option template refresh interval.
	<1-216000>	Specifies the NetFlow Generation template timeout in seconds. The default is 1800.
apps netflow exporter alias <alias> transport		Specifies the NetFlow Generation Exporter Transport Protocol.

Command	Argument	Description
	udp	Uses the NetFlow Generation UDP Transport Protocol. This is the default. The default port for syslog (for CEF format) is 514.
	<port>	Specifies the port on which the NetFlow Generation Collector is listening. The default port for NetFlow is 2055.
apps netflow exporter alias <alias> ttl		Specifies the NetFlow Generation Time-To-Live (TTL) value.
	<1-255>	Specifies the NetFlow Generation Time-To-Live value in seconds. The default is 64.
apps netflow monitor		Configures a NetFlow Generation Monitor.
	alias	Configures an alias for the NetFlow Generation Monitor.
apps netflow monitor alias	<alias>	Specifies the NetFlow Monitor alias.
apps netflow monitor alias <alias>	<cr>	Enters *NetFlow Generation Monitor Mode* to configure a NetFlow Generation Monitor.
	cache	Configures NetFlow Generation cache parameters.
	description	Specifies a description for the NetFlow Generation Monitor.
	port-list	Configures the monitor to scan specific ports for SSL. Use with NetFlow SSL metadata.
	record	Associates a NetFlow Generation Record to the NetFlow Generation Monitor.
	sampling	Configures NetFlow Generation Monitor sampling parameters.
apps netflow monitor alias <alias> cache		Specifies cache parameters for the NetFlow Generation Monitor.
	timeout	Specifies a timeout for the entries in the NetFlow Generation cache.
apps netflow monitor alias <alias> cache timeout		Configures the monitor cache timeout.
	active	Specifies the active NetFlow Generation timeout in seconds.
	event	Specifies that the NetFlow Generation Record is generated and exported in the NetFlow Generation cache on an event.

Command	Argument	Description
	inactive	Specifies the inactive NetFlow Generation timeout in seconds.
apps netflow monitor alias <alias> cache timeout active		Configures monitor cache timeout active.
	<1-604800>	Specifies the active NetFlow Generation timeout value in seconds. The default is 1800.
apps netflow monitor alias <alias> cache timeout event		Configures monitor cache timeout event.
	none	Configures a monitor cache timeout event of **none**.
	transaction-end	Specifies that the NetFlow Generation Record is generated and exported in the NetFlow Generation Cache at the end of a transaction.
apps netflow monitor alias <alias> cache timeout inactive		Configures monitor cache timeout inactive.
	<1-604800>	Specifies the inactive NetFlow Generation timeout value in seconds. The default is 15.
apps netflow monitor alias <alias> description	<string>	Specifies a description for the NetFlow Generation Monitor.
apps netflow monitor alias <alias> port-list add	<0 to 65535>	Adds specific ports to scan for SSL. List up to 10 ports to attach to the monitor. Use commas to separate the ports in the list. For example, 443,993,1000.
apps netflow monitor alias <alias> port-list all		Adds all ports to scan for SSL.
apps netflow monitor alias <alias> port-list delete	<0 to 65535>	Deletes specific ports to scan for SSL. List up to 10 ports to delete from the monitor. Use commas to separate the ports in the list. For example, 993,636.
apps netflow monitor alias		Specifies the following SSL ports to scan for SSL: • IMAP_SSL_PORT 993

Command	Argument	Description
<alias> port-list well-known-ports		• POP3_SSL_PORT 995 • SMTP_SSL_PORT 465 • LDAP_SSL_PORT 636 • NNTP_SSL_PORT 563 • HTTP_SSL_PORT 443
apps netflow monitor alias <alias> record		Adds or deletes a NetFlow record to or from a monitor.
	add	Adds one or more records to a monitor, up to 5.
apps netflow monitor alias <alias> record add	<monitor record>	Specifies a name of a monitor record.
	predefined netflow_v5_ record	Specifies a predefined NetFlow monitor record, a V5 fixed record template.
apps netflow monitor alias <alias> record delete	delete <all | record-id>	Deletes all records associated with the monitor or deletes a specific record from a monitor using the record identifier.
apps netflow monitor alias <alias> sampling		Enables sampling or defines the sampling rates for a NetFlow monitor.
	set	Enables sampling for a NetFlow monitor and specifies the type of sampling to be performed: • multi-rate • no-sampling • single-rate
apps netflow monitor alias <alias> sampling set	multi-rate	Enables multi-rate sampling for a NetFlow monitor. Multi-rate sampling can be applied to any record.
	no-sampling	Disables sampling for a NetFlow monitor.
	single-rate	Enables single-rate sampling for a NetFlow monitor. Single-rate applies to all records.
apps netflow monitor alias <alias> sampling	1 in <10-16000>	Defines the sampling rate for single-rate sampling by specifying a number for 1 in N, where N is the packet count from 10 to 16000.

Command	Argument	Description
single-rate		
apps netflow record		Configures a NetFlow Generation Record template.
	alias	Configures an alias for a NetFlow Generation Record.
apps netflow record alias		Specifies an alias name for a NetFlow Generation Record.
	<alias>	Specifies a NetFlow record alias.
apps netflow record alias <alias>		Specifies the NetFlow Generation Record alias parameters.
	<cr>	Enters *NetFlow Generation Record Mode* to configure a NetFlow Generation Record template.
	collect	Configures a non-key field for the NetFlow Generation Record.
	description	Specifies a description for the NetFlow Generation Record (optional).
	export-blank-pen	Configures the export of a blank pen record that contains a mix of private enterprise elements and non-private enterprise elements, however during runtime, the private enterprise elements are empty. The options are as follows: • **yes**—Exports the blank or empty PEN records to the collector. • **no**—Does not export the blank or empty PEN records to the collector.
	exporter	Assigns an exporter to a NetFlow record.
	match	Configures a key field for the NetFlow Generation Record.
	netflow-version	Specifies the NetFlow Generation Record version.
	sampling	Adds or deletes sampling for a NetFlow record. 1 in 1 (no sampling) is the default.
apps netflow record alias <alias> collect		Specifies the NetFlow Generation non-key fields of the NetFlow Generation Record.
	add	Adds a new NetFlow Generation Collect non-key field.
	delete	Deletes an existing NetFlow Generation collect non-key field.
apps netflow		Specifies the collect type and its parameters.

Command	Argument	Description
record alias <alias> collect add <collect_ type> <parameters>		
apps netflow record alias <alias> collect add counter		Adds a new NetFlow Generation Collect counter field.
apps netflow record alias <alias> collect add datalink		Adds a new NetFlow Generation Collect datalink field.
apps netflow record alias <alias> collect add flow	end-reason	Adds a new NetFlow Generation Collect flow field.
apps netflow record alias <alias> collect add interface input name	input name [width <1-32>]	Specifies the interface name as follows: • **width**—Specifies an optional parameter that indicates the size of the field, from 1 to 32 bytes. If not specified, the width of the interface name will be a maximum of 32 bytes. In the NetFlow record, the collect field for the interface input name includes the interface ID in the format <box ID>/<slot ID>/<port ID>, for example, 1/1/x1, as well as the alias, if there is an alias associated with the interface. The total number of characters for the interface ID and alias is 128.
apps netflow record alias <alias> collect add interface input physical | output physical	<input | output> physical [width <2 | 4>]	Specifies the recording interface (ingress and/or egress) as one of the fields to be sent in the NetFlow record as follows: • **input** and/or **output**—Specifies one or both ingress and egress interfaces as collect fields. • **width**—Specifies an optional parameter that indicates the size of the field. The valid values are as follows: o IPFIX: 4 bytes. o V9: 2 or 4 bytes. The default is 2. Do not use the default of 2 for v9. Specify a width of 4 to match the actual interface port ID width, which is 4 bytes.
apps netflow record alias alias-name collect add exporter ipv4-addr		Adds new switch or router management ipv4 address.

Command	Argument	Description
apps netflow record alias alias-name collect add exporter ipv6-addr		Add new switch or router management ipv6 address.
apps netflow record alias <alias> collect add ipv4		Adds a new NetFlow Generation Collect ipv4 field.
apps netflow record alias <alias> collect add ipv6		Adds a new NetFlow Generation Collect ipv6 field.
apps netflow record alias <alias> collect add private	**pen <pen name> dns <additional-class [number-of-collects <1-10>] | additional-class-text [number-of-collects <1-10>] |additional-name [number-of-collects <1-10>] | additional-rd-length [number-of-collects <1-10>] | additional-rdata [number-of-collects <1-10> | width <1-128>] | additional-ttl [number-of-collects <1-10>] | additional-type [number-of-collects <1-10>] | additional-type-text [number-of-collects <1-10>] | an-count | ar-count |authority-class [number-of-collects <1-10>] |**	Specifies the private enterprise name (pen) for capturing packets containing Domain Name Service (DNS) information as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **number-of-collects**—Specifies an optional parameter that indicates the number of instances of elements that can be collected for a DNS request. The default value is 1. The range is from 1 to 10. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 128 bytes. The default is 64. • **additional-class**—Specifies the additional class containing one of the RR class codes. • **additional-class-text**—Specifies the text string of the hexadecimal value of the additional class containing one of the RR class codes. • **additional-name**—Specifies the domain name in the additional records section. • **additional-rd-length**—Specifies the length of the rdata field in the additional records section. • **additional-rdata**—Specifies the length of the rdata field in the additional records section. • **additional-ttl**—Specifies the time-to-live (TTL), which is the time interval in seconds that the record is cached in the additional records section. • **additional-type**—Specifies the additional type containing one of the RR type codes. • **additional-type-text**—Specifies the text string of the hexadecimal value of the additional type containing one of

Command	Argument	Description
		the RR type codes. • **an-count**—Specifies the number of resource records in the answer section. • **ar-count**—Specifies the number of resource records in the additional records section. • **authority-class**—Specifies the authority class containing one of the RR class codes.
apps netflow record alias <alias> collect add private (continued)	**authority-class-text [number-of-collects <1-10>] | authority-name [number-of-collects <1-10>] | authority-rd-length [number-of-collects <1-10>] | authority-rdata [number-of-collects <1-10> | width <1-128>] | authority-ttl [number-of-collects <1-10>] | authority-type [number-of-collects <1-10>] | authority-type-text [number-of-collects <1-10>] | bits | identifier | ns-count | op-code | qd-count | query-class [number-of-collects <1-10>] | query-class-text [number-of-collects <1-10>] | query-name [number-of-collects <1-10>] | query-type [number-of-collects <1-10>] |**	• **authority-class-text**—Specifies the text string of the hexadecimal value of the authority class containing one of the RR class codes. • **authority-name**—Specifies the domain name in the authority section. • **authority-rd-length**—Specifies the length of the rdata field in the authority section. • **authority-rdata**—Specifies the content that describes the resource in the authority section. • **authority-ttl**—Specifies the time-to-live (TTL), which is the time interval in seconds that the record is cached in the authority section. • **authority-type**—Specifies the authority type containing one of the RR type codes. • **authority-type-text**—Specifies the text string of the hexadecimal value of the authority type containing one of the RR type codes. • **bits**—Specifies the variable length of a bit map. • **identifier**—Specifies an identifier generated by the device that creates the DNS query and is copied by the server into the response so it can be used by that device to match that query to the corresponding reply received from the DNS server. • **ns-count**—Specifies the number of the name server (NS) resource records in the authority records section. • **op-code**—Specifies the query type. • **qd-count**—Specifies the number of entries in the question section. • **query-class**—Specifies the query format containing one of the RR class codes. • **query-class-text**—Specifies the text string of the hexadecimal value of the query class containing one of the RR type codes. • **query-name**—Specifies the domain name requested in the query (maximum 64 bytes). • **query-type**—Specifies the query format containing one of the RR type codes.

Command	Argument	Description
apps netflow record alias <alias> collect add private (continued)	**query-type-text [number-of-collects <1-10>] | response-class [number-of-collects <1-10>] | response-class-text [number-of-collects <1-10>] | response-code | response-ipv4-addr [number-of-collects <1-10>] |response-ipv4-addr-text [number-of-collects <1-10>] |response-ipv6-addr [number-of-collects <1-10>] | response-ipv6-addr-text [number-of-collects <1-10>]response-name [number-of-collects <1-10>] | response-rd-length [number-of-collects <1-10>] | response-rdata [number-of-collects <1-10> | width <1-128>] | response-ttl [number-of-collects <1-10>] | response-type [number-of-collects <1-10>] |response-type-text [number-of-collects <1-10>]>**	• **query-type-text**—Specifies the text string of the hexadecimal value of the query format containing one of the RR type codes. • **response-class**—Specifies the response format containing one of the RR class codes. • **response-class-text**—Specifies the text string of the hexadecimal value of the response format containing one of the RR class codes. • **response-code**—Specifies the type of the response. • **response-ipv4-addr**—Specifies the IPv4 address in the response if the response type host and class are Internet/IPv4. • **response-ipv4-addr-text**—Specifies the text string of the hexadecimal value of the IPv4 address in the response if the response type host and class are Internet/IPv4. • **response-ipv6-addr**—Specifies the IPv6 address in the response if the response type host and class are Internet/IPv6. • **response-ipv6-addr-text**—Specifies the text string of the hexadecimal value of the IPv6 address in the response if the response type host and class are Internet/IPv6. • **response-name**—Specifies the domain name in the response (maximum 64 bytes). • **response-rd-length**—Specifies the length of the rdata field in the response data field. • **response-rdata**—Specifies the content that describes the resource in the response data field. • **response-ttl**—Specifies the time-to-live (TTL), which is the time interval in seconds that the record is cached. • **response-type**—Specifies the response type containing one of the RR Type codes. • **response-type-text**—Specifies the text string of the hexadecimal value of the response type containing one of the RR Type codes.
apps netflow record alias <alias> collect add private	pen <pen name> http response-code	Specifies the private enterprise name (pen) for capturing HTTP response codes as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **response-code**—Captures any packet with an HTTP response code embedded in it. For IPFIX only.
apps netflow record alias <alias> collect add private	pen <pen name> http url [width <1-250>]	Specifies the private enterprise name (pen) for capturing packet URLs as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **url**—Captures any packet with a URL embedded in it. For

Command	Argument	Description
		IPFIX only. • **width**—Specifies an optional parameter that indicates the maximum URL length that is allowed in the data record, from 1 to 250 bytes. If not specified, the URL will be a maximum of 128 bytes.
apps netflow record alias <alias> collect add private	pen <pen name> http user-agent [width <1-250>]	Specifies the private enterprise name (pen) for capturing user agents as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **user-agent**—Gathers information about the user agent involved in the packet transfer. The user agent appears in the HTTP request header. It is a variable that will be filled in by the browser. • **width**—Specifies an optional parameter that indicates the maximum user agent length that is allowed in the data record, from 1 to 250 bytes. If not specified, the user agent length has a default of 150 bytes.
apps netflow record alias <alias> collect add private	pen <pen name> ssl certificate <issuer [width <1-250>] | issuerCommonName [width <1-64>] | serialNumber | serialNumber-text | signatureAlgorithm |signatureAlgorithm-text | subject [width <1-250>] | subjectAlgorithm | subjectAlgorithm-text |	Specifies the private enterprise name (pen) for capturing SSL certificate metadata as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **issuer**—Specifies the certificate issuer. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 250 bytes. The default is 128. • **issuerCommonName**—Specifies the certificate issuer common name, which is a subset of **issuer**. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 64 bytes. The default is 32. • **serialNumber**—Specifies the unique number for each certificate issued by a given CA. • **serialNumber-text**—Specifies the text string of the hexadecimal value of the unique number for each certificate issued by a given CA. • **signatureAlgorithm**—Specifies the identifier for the cryptographic algorithm used by the CA to sign the certificate. • **signatureAlgorithm-text**—Specifies the text string of the hexadecimal value of the identifier for the cryptographic algorithm used by the CA to sign the certificate. • **subject**—Specifies the certificate subject. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 250 bytes. The default

Command	Argument	Description
		is 128. • **subjectAlgorithm**—Specifies the subject public key algorithm used, such as RSA or DSA. • **subjectAlgorithm-text**—Specifies the text string of the hexadecimal value of the subject public key algorithm used, such as RSA or DSA.
apps netflow record alias <alias> collect add private (continued)	subjectAltName [width <1-64>] | subjectCommonName | [width <1-64>] | subjectKeySize | validNotAfter | validNotAfter-text | validNotBefore | validNotBefore-text>	• **subjectAltName**—Specifies the subject alternative name, which allows identities to be bound to the **subject**. The first **subjectAltName** present in the certificate is collected. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 64 bytes. The default is 32. • **subjectCommonName**—Specifies the certificate subject common name, which is a subset of **subject**. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 64 bytes. The default is 32. • **subjectKeySize**—Specifies the subject public key size. • **validNotAfter**—Specifies the date on which the certificate validity period ends. The format is YYMMDDHHMMSSZ, where Z is Zulu time (GMT). • **validNotAftertext**—Specifies the text string of the date on which the certificate validity period ends. The format is MMM DD HH:SS YYYY GMT. • **validNotBefore**—Specifies the text string of the date on which the certificate validity period begins. The format is YYMMDDHHMMSSZ, where Z is Zulu time (GMT). • **validNotBefore-text**—Specifies the text string of the date on which the certificate validity period begins. The format is MMM DD HH:SS YYYY GMT.
apps netflow record alias <alias> collect add private	pen <pen name> ssl server <cipher | cipher-text | compressionMethod | nameIndication [width <1-64>] | sessionId | version | version-text>	Specifies the private enterprise name (pen) for capturing SSL server metadata as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **cipher**—Specifies the cipher that the server agreed to use for that session. • **cipher-text**—Specifies the text string of the hexadecimal value of the cipher that the server agreed to use for that session. • **compressionMethod**—Specifies the server compression method, which is typically NULL. • **nameIndication**—Specifies the extension to the TLS protocol by which a client indicates the hostname to which it is attempting to connect at the start of the handshaking

Command	Argument	Description
		process. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 64 bytes. The default is 32. • **sessionId**—Specifies the session identifier, generated by a server, which identifies a particular session. • **version**—Specifies the version of SSL. • **version-text**—Specifies the text string of the hexadecimal value of the version of SSL.
apps netflow record alias <alias> collect add private	pen <pen name> ssl server <cipher | cipher-text | compressionMethod | nameIndication [width <1-64>] | sessionId | version | version-text>	Specifies the private enterprise name (pen) for capturing SSL server metadata as follows: • **pen <pen name>**—Specifies a pen name. The only valid pen name is gigamon. • **cipher**—Specifies the cipher that the server agreed to use for that session. • **cipher-text**—Specifies the text string of the hexadecimal value of the cipher that the server agreed to use for that session. • **compressionMethod**—Specifies the server compression method, which is typically NULL. • **nameIndication**—Specifies the extension to the TLS protocol by which a client indicates the hostname to which it is attempting to connect at the start of the handshaking process. • **width**—Specifies an optional parameter that indicates the maximum length of the field, from 1 to 64 bytes. The default is 32. • **sessionId**—Specifies the session identifier, generated by a server, which identifies a particular session. • **version**—Specifies the version of SSL. • **version-text**—Specifies the text string of the hexadecimal value of the version of SSL.
apps netflow record alias <alias> collect add timestamp	flow-start-sec | flow-end-sec | flow-start-msec | flow-end-msec | sys-uptime <first | last>	Adds a new NetFlow Generation Collect timestamp field.
apps netflow record alias <alias> collect add transport		Adds a new NetFlow Generation Collect transport field.
apps netflow		Configures record collect delete.

Command	Argument	Description
record alias <alias> collect delete		
	all	Deletes all NetFlow Generation collect non-key fields with an associated NetFlow Generation Record.
	collect-id	Deletes NetFlow Generation collect non-key fields corresponding to a particular Collect ID.
apps netflow record alias <alias> collect delete collect-id		Deletes collect key field corresponding to a Collect ID.
	<integer>	Specifies the collect ID.
apps netflow record alias <alias> description		Specifies a description for the NetFlow Generation Record.
	<string>	Specifies a description for the NetFlow Generation Record.
apps netflow record alias <alias> export-blank-pen		Specifies whether or not to export a NetFlow record when there is a mix of private and non-private elements in the record.
	no	Does not export the NetFlow record when there is a mix of private and non-private elements in the record.
	yes	Exports the NetFlow record when there is a mix of private and non-private elements in the record.
apps netflow record alias <alias> exporter		Adds an exporter to a NetFlow record, or removes an exporter from a NetFlow record.
	add <record exporter>	Adds an exporter to a NetFlow record.
	delete <all | exporter-id <exporter-id>	Removes an exporter from a NetFlow record or removes all exporters.
apps netflow record alias <alias> match		Specifies key fields for the NetFlow Generation Record.
	add	Adds a NetFlow Generation new match key field.
	delete	Deletes an existing NetFlow Generation match key field.

Command	Argument	Description
apps netflow record alias <alias> match add <match_ type> <parameters>		Specifies the NetFlow Generation match type and its associated parameters.
apps netflow record alias <alias> match add datalink		Adds a new NetFlow Generation Match datalink field.
apps netflow record alias <alias> match add interface	input physical [width <2 | 4>]	Specifies the input interface as one of the key fields for flow identification as follows: • **input**—Denotes using the packet ingress interface as key field. • **width**—Specifies an optional parameter that indicates the size of the field. The valid values are as follows: o IPFIX: 4 bytes. o V9: 2 or 4 bytes. The default is 2. Do not use the default of 2 for v9. Specify a width of 4 to match the actual interface port ID width, which is 4 bytes.
apps netflow record alias <alias> match add ipv4		Adds a new NetFlow Generation Match ipv4 field.
apps netflow record alias <alias> match add ipv6		Adds a new NetFlow Generation Match ipv6 field.
apps netflow record alias <alias> match add transport		Adds a new NetFlow Generation Match transport field.
apps netflow record alias <alias> match delete		Configures record match delete.
	all	Deletes all NetFlow Generation Match key fields for a particular NetFlow Generation Flow Record.
	match-id	Deletes the NetFlow Generation match key field corresponding to a particular Match ID.
apps netflow		Deletes match key field corresponding to Match ID.

Command	Argument	Description
record alias <alias> match delete match-id		
	<integer>	Specifies a match ID.
apps netflow record alias <alias> netflow-version		Specifies a version for the NetFlow Generation Record.
	ipfix	Specifies NetFlow Generation Version IPFIX.
	netflow-v9	Specifies NetFlow Generation version 9. This is the default.
apps netflow record alias <alias> sampling		Specifies a sampling rate or disables sampling on a NetFlow record.
	delete	Disables sampling on a NetFlow record.
	set	Specifies the sampling rate for a NetFlow record as 1 in N, where N is a number from 1 to 16000.
gsparams gsgroup <alias>		Associates a NetFlow Generation Monitor to a specified GigaSMART group.
	netflow-monitor	Enables a NetFlow Generation Monitor on a GigaSMART group.
gsparams gsgroup <alias> netflow-monitor		Configures NetFlow Monitor.
	add <Monitor name>	Adds a NetFlow Generation Monitor.
	delete	Deletes a NetFlow Generation Monitor.
gsop alias <alias> flow-ops		Enables flow processing.
	netflow	Enables NetFlow Generation.

Related Commands

The following table summarizes other commands related to the **apps netflow** command:

Task	Command
Displays general NetFlow information.	**# show apps netflow**
Displays NetFlow exporters.	**# show apps netflow exporter**
Displays NetFlow exporter for a specified alias.	**# show apps netflow exporter alias exp1**
Displays all NetFlow exporters configured.	**# show apps netflow exporter all**
Displays NetFlow exporter statistics.	**# show apps netflow exporter stats**
Displays statistics for a specified NetFlow exporter.	**# show apps netflow exporter stats alias exp1**
Displays statistics for all NetFlow exporters.	**# show apps netflow exporter stats all**
Displays NetFlow monitors.	**# show apps netflow monitor**
Displays NetFlow monitor for a specified alias.	**# show apps netflow monitor alias mon1**
Displays all NetFlow monitors configured.	**# show apps netflow monitor all**
Displays NetFlow monitor statistics.	**# show apps netflow monitor stats**
Displays statistics for a specified NetFlow monitor.	**# show apps netflow monitor stats alias mon1**
Displays statistics for all NetFlow monitors.	**# show apps netflow monitor stats all**
Displays NetFlow port ID.	**# show apps netflow port-id**
Displays NetFlow records.	**# show apps netflow record**
Displays NetFlow record for a specified alias.	**# show apps netflow record alias rec1**
Displays all NetFlow records configured.	**# show apps netflow record all**
Deletes a specified NetFlow exporter.	**(config) # no apps netflow exporter alias exp1**
Disables SNMP for a specified NetFlow exporter.	**(config) # no apps netflow exporter alias exp1 snmp enable**
Disables SNMP for a specified NetFlow exporter.	**(config) # apps netflow exporter alias exp1 no snmp enable**
Deletes all NetFlow exporters.	**(config) # no apps netflow exporter all**

Task	Command
Deletes a specified NetFlow monitor.	**(config) # no apps netflow monitor alias mon1**
Deletes all NetFlow monitors.	**(config) # no apps netflow monitor all**
Deletes a specified NetFlow record.	**(config) # no apps netflow record alias rec1**
Deletes the predefined NetFlow v5 record.	**(config) # no apps netflow record alias predefined_ netflow_v5_record**
Deletes all NetFlow records.	**(config) # no apps netflow record all**

apps regrex-profile

Use the **apps regrex-profile** command to define a set of matching pattern consisting of regular expression.

The **apps regrex-profile**command has the following syntax:

```
apps regrex-profile alias <name>
   pattern add <"regular expression">
   pattern delete <pattern-id>
exit
```

The following table describes the arguments for the **apps regrex-profile**command:

Argument	Description
alias <name>	Specifies the regrex-profile alias. You can create a maximum of 10 regrex profiles.
pattern add <"regular expression">	Specifies Perl compatible regular expression pattern. You can create a maximum of 50 patterns per profile.
pattern delete <pattern-id>	Removes configured regular expression pattern.

apps sip-whitelist

Required Command-Line Mode = Configure

Use the **apps sip-whitelist** command to configure SIP forward listing.

The **apps sip-whitelist**command has the following syntax:

```
apps sip-whitelist alias <SIP whitelist file alias>
  add callerid <caller/callee ID>
  add id-range <id-range>
  add ip-addr <ip-address>
```

create
delete <all | callerid <caller ID>>
destroy
fetch <add | delete> <URL for a SIP whitelist file>

The following table describes the arguments for the **apps sip-whitelist** command:

Argument	Description
sip-whitelist alias <SIP whitelist file alias>	Specifies an alias of the forward list file.
add callerid <caller ID>	Adds a single caller ID entry to a forward list. Specify up to 64 alphanumeric characters. The supported characters include: • lower case alphabetic, a-z • upper case alphabetic, A-Z • numeric, 0-9 • hyphen, - • underscore, _ • period, . • exclamation, ! • tilde, ~ • open bracket, (• close bracket,) • asterisk, * • ampersand, & • equals sign, = • plus sign, + • dollar sign, $ • comma, , • semi-colon, ; • question mark, ? • forward slash, / • at sign, @ For example: **(config) # apps sip-whitelist alias sip-scp add callerid 302701237777777**
add id-range <id-range>	Adds the range of caller-ids to the forward list entries from the given start range to the end range . The start and the end range must contain a maximum of 64 numeric characters. The supported characters include: • numeric, 0-9 • range, .. For example: **(config) # apps sip-whitelist alias sip-scp add id-range**

Argument	Description
	123456700..123456750
add ip-addr <ip-address>	Adds a valid IPv4/IPv6 address to the forward list entries. IP address must be a valid single: • IPv4 address • IPv6 address For example: **(config) # apps sip-whitelist alias sip-scp add ip-addr 192.168.1.1**
create	Creates a new forward list. For example: **(config) # apps sip-whitelist alias sip-scp create** To create a forward list, refer to How to Create a Forward List.
delete <all | callerid <caller ID>>	Specifies actions for **delete** as follows: • **all**—Deletes all forward list entries. This deletes all caller ID entries, up to 500,000. • **callerid**—Deletes a single caller ID entry from a forward list. When using **delete all** to delete a forward list, unlike **destroy**, you do not have to delete the forward list maps, the GigaSMART operation, or disassociate the GigaSMART group from the forward list. Examples: **(config) # apps sip-whitelist alias sip-scp delete callerid 302701237777777** **(config) # apps sip-whitelist alias sip-scp delete all**
destroy	Destroys a forward list. For example: **(config) # apps sip-whitelist alias sip-scp destroy** When using **destroy** to delete a forward list, unlike **delete all**, you must first delete the forward list maps, the GigaSMART operation, and disassociate the GigaSMART group from the forward list before deleting the forward list. For the procedure to destroy the forward list, refer to How to Destroy a Forward List.
fetch <add | delete> <URL for a SIP whitelist file>	Specifies actions for **fetch** as follows: • **add**—Downloads a forward list file from a specified URL and path. Use this parameter to add up to 20,000 caller IDs. • **delete**—Deletes the caller ID entries, located in the forward list file at the specified URL and path, from the forward list on the node. Use this option to delete up to 20,000 caller IDs. • For both **add** and **delete**, forward list files must adhere to the following: • The caller IDs in forward list files must be distinct entries, with one caller ID on each line of a file. • In a forward list file, use only the carriage return (newline) to separate caller ID entries. Do not use any characters, such as commas or colons, to separate caller ID entries in forward list files. • Each forward list file can contain a maximum of 20,000 entries.

Argument	Description
	• Forward list files must have a filename with a .txt suffix. To fetch a specified forward list file from a location, use one of the following formats: • http://IPaddress/path/filename.txt • scp://username:password@IPaddress:/path/filename.txt For SIP forward listing in a cluster, only fetch the forward list to the leader in the cluster. On member nodes, **fetch** is not available. Examples: **(config) # apps sip-whitelist alias sip-scp fetch add http://1.1.1.1/tftp/temp/MyIDs1.txt** **(config) # apps sip-whitelist alias sip-scp2 fetch add scp://user1:mypw@1.1.1.1:/home/temp/C_ID_file1.txt** **(config) # apps sip-whitelist alias sip-scp fetch delete http://1.1.1.1/tftp/temp/MyIDstoDelete.txt** **(config) # apps sip-whitelist alias sip-scp2 fetch delete scp://user1:mypw@1.1.1.1:/home/temp/C_ID_delfile.txt**

How to Create a Forward List

To create a forward list, use the following CLI command sequence:

Task	Command
Create the forward list.	**(config) # apps sip-whitelist alias sip-scp create**
Associate the GigaSMART group to the forward list.	**(config) # gsparams gsgroup gsg1 sip-whitelist add sip-scp**

Task	Command
Configure the GigaSMART operation.	**(config) #** gsop alias **sip_wl1** flow-ops sip-whitelist lb app sip metric hashing key caller-id port-list gsg1
Add single entries to the whitelist.or Fetch and download forward list files.	**(config) # apps sip-whitelist alias sip-scp1 add callerid 302701237777777** **(config) # apps sip-whitelist alias sip-scp1 add callerid 302701237777778** **(config) # apps sip-whitelist alias sip-scp1 fetch add http://1.1.1.1/tftp/temp/whitelist1.txt** **(config) # apps sip-whitelist alias sip-scp1 fetch add http://1.1.1.1/tftp/temp/whitelist2.txt**
Create a second level map, the forward list map. When the map configuration is complete, the forward list will take effect. **Note**: The SIP forward list map does not have any rules.	**(config) # map alias SIP-WL-S11** **(config map alias SIP-WL-S11) # type secondLevel flowWhitelist-sip** **(config map alias SIP-WL-S11) # from vp1** **(config map alias SIP-WL-S11) # use gsop sip_wl1** **(config map alias SIP-WL-S11) # to pg-wl-1** **(config map alias SIP-WL-S11) # exit** **(config) #**

How to Destroy a Forward List

To destroy the entire forward list, use the following CLI command sequence:

Task	Command
Delete a forward list map.	**(config) # no map alias SIP-WL-S11**
Delete the GigaSMART operation.	**(config) # no gsop alias sip_wl1**
Disassociate the GigaSMART group from the forward list. (You do not need to delete the **gsgroup**.)	**(config) # gsparams gsgroup gsg1 sip-whitelist delete**
Destroy () the entire forward list.	**(config) # apps sip-whitelist alias sip-scp1 destroy**

Related Commands

The following table summarizes other commands related to the **apps sip-whitelist** command:

Task	Command
Displays a particular caller ID associated with the GigaSMART group.	**# show gsgroup sip-whitelist alias gsg1 caller-id 302701237777777**
Displays the SIP forward list entry count.	**# show apps sip-whitelist alias sip-scp count**

apps split-dns

You can choose to configure separate DNS servers for internal and external networks to ensure better security and privacy management. If you choose to add an internal DNS server, you must create a split-DNS profile, add a collector DNS server for external networks, and then add the required rules for the profile.

Use the **apps split-dns** command to configure split-DNS profile for each GigaSMART engine port. You can configure up to a maximum of five split-DNS profiles for a device, however, you can enable only one split-DNS profile for a GigaSMART engine port. You must configure only one collector DNS server for a split-DNS profile. You can configure up to a maximum of 100 rules in a split-DNS profile. When you configure the rules, keep in mind the following:

- Duplicate rules are not allowed.
- Multicast or broadcast IP addresses are not allowed.
- The valid format for domain name is *.name.org, xxx.name.com, or name.com.

The **apps split-dns** command has the following syntax:

apps split-dns profile alias <alias>
collector add dns <ip-address>
rule add dns <ip-address> domain <domain-name>
collector edit dns <ip-address>
rule edit id <rule id> dns <ip-address>| domain <domain name>
rule delete id <rule id>| all
exit

The following table describes the arguments for the **apps split-dns** command:

Argument	Description
apps split-dns profile alias <alias>	Specifies an alias for the split-DNS profile. Example: **(config) # apps split-dns profile alias splitdns1**
collector add dns <ip-address>	Configures a collector DNS server for the split-DNS profile. Example: **(config) # collector add dns 10.115.181.228**

Argument	Description
rule add dns <ip-address> domain <domain-name>	Configures a rule with DNS server IP address and domain name for the split-DNS profile. Example: **(config) # rule add dns 11.22.33.44 domain *.gigamon.com**
collector edit dns <ip-address>	Edits the IP address of the collector DNS server. Example: **(config) # collector edit dns 10.115.181.229**
rule edit id <rule id> dns <ip-address>| domain <domain name>	Edits the DNS server IP address or the domain name configured for the rule. **NOTE:** You cannot edit the rules that are not configured for the specified split-DNS profile. Examples: **(config) # rule edit id 2 dns 10.20.30.40** **(config) # rule edit id 2 domain *.yahoo.com**
rule delete id <rule id>| all	Deletes the specified rule ID or all the rules configured for the split-DNS profile. **Note**: You cannot delete the rules that are not configured for the specified split-DNS profile. Examples: **(config) # rule delete id 2** **(config) # rule delete all**

Related Commands

The following table summarizes other commands related to the **apps split-dns**command:

Task	Command
Displays details such as the collector DNS server and rules configured for the specified split-DNS profile.	**# show apps split-dns profile alias <alias>**
Displays the details of all the split-DNS profiles configured for the device.	**# show apps split-dns profile all**

Task	Command
Deletes the specified split-DNS profile. **Note**: Before deleting the split-DNS profile, ensure that you disable the profile from the GigaSMART engine port. To disable the profile, you must delete the GigaSMART engine port configurations. Refer to gigasmart.	**# no apps split-dns profile alias <alias>**
Deletes all the split-DNS profiles configured for the device. **Note**: Before deleting the split-DNS profiles, ensure that you disable the profiles from the GigaSMART engine ports. To disable the profile, you must delete the GigaSMART engine port configurations. Refer to gigasmart.	**# no apps split-dns profile all**

apps ssl

Use the **apps ssl** command to configure Secure Sockets Layer (SSL) parameters for Passive SSL decryption.

The **apps ssl** command has the following syntax:

```
apps ssl   key alias <alias>
      comment <comment>
      download type
         pkcs12 <url <download URL>> [password <password>]
        private-key <key-str <key string> | url <download URL>>
   keychain password <password> <confirm password> | <password> | [reset] <password>
<confirm password>
   service alias <alias>
      default-service
      server-ip <IP address> [server-port <port number> | any]
```

You must have an admin level role to execute these commands.

The following table describes the arguments for the **apps ssl** command:

Argument	Description
key alias <alias> **comment <comment>**	Adds a comment to an existing SSL private key. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # apps ssl key alias key1 comment “This is a comment”**
key alias <alias> **download type** **pkcs12 <url**	Downloads SSL key types as follows: • **pkcs12**—Specifies a PKCS12 file containing the private key and the certificate.

Argument	Description
<download URL>> [password <password>] private-key <key-str <key string> | url <download URL>>	• **private-key**—Specifies a private key. The parameters are as follows: • **url**—Specifies the download URL for either PKCS12 or private key. • **password**—Specifies an optional password for PKCS12. If a password is not specified after the **password** keyword, you will be prompted for it. • **key-str**—Specifies the SSL key PEM file by providing a key string for a private key. Enclose the key string in double quotation marks. Examples: (config) # apps ssl key alias key1 download type private-key url https://keyserver.domain.com/path/keyfile.pem **(config) # apps ssl key alias key2 download type pkcs12** url sftp://test:mytest@10.10.10.10/home/test/ssldecrypt/keys/srv1k.pfx **(config) # apps ssl key alias key3 download type private-key key-str "-----BEGIN RSA PRIVATE KEY----- ...-----END RSA PRIVATE KEY-----"** The download URL specifies an SSL private key or PKCS12 container. The supported formats for download are HTTP, HTTPS, FTP, TFTP, SCP, and SFTP. Using a secure protocol, such as HTTPS is recommended. With PKCS12, the key will be converted to PEM format, the certificate will be verified, then the key will be added to the keychain. Once the key is added in PEM format to the keychain, no checks will be performed to verify if it has expired. For more information on keys, refer to the "*GigaSMART SSL Decryption for Out-of-Band Tools*" section in the *GigaVUE Fabric Management Guide*.
keychain password <password> <confirm password>	Creates an SSL keychain password. Use this command when no keys have been installed on the node, for example: **(config) # apps ssl keychain password**Creating a new password for ssl keychain: **Password: ********* Confirm: ********* The password is used to encrypt all private keys uploaded to the node. Only strong passwords can be configured. A strong password has at least 10 characters and at least three of the following: • uppercase letters • lowercase letters • numbers • special characters **Note**: The password is not saved on the node.
keychain password <password>	Prompts for the SSL keychain password. When keys are installed on the node, you will be prompted to verify the password after any node reboot when you enter configure terminal mode, for example: **(config)# configure terminal**

Argument	Description
	(config) # apps ssl keychain password required Please enter ssl keychain password: **Password: ***********
keychain password [reset] <password> <confirm password>	Resets an SSL keychain password. When keys are installed on the node, a warning is displayed. **Note**: Resetting the password revokes all existing private keys. For example: **(config) # apps ssl keychain password resetWARNING: Password is already set. Reset password will revoke all existing private keys.** **Password: ********* Confirm: ***********
service alias <alias> default-service	Specifies a default SSL service. The default service matches any IP address mapped to a valid key. **Note**: There can only be one default service defined. For example: **(config) # apps ssl service service1 default-service** Refer to gsparams for information on mapping the default service to a private key.
service alias <alias> server-ip <IP address> [server-port <port number> | any]	Adds a server IP address to a specified service and optionally, adds a server port number. For example: **(config) # apps ssl service service1 server-ip 1.1.1.1 server-port 443** A server port number of **any** means any port. The key applies to the whole IP address, not just to a specified port. For example: **(config) # apps ssl service service1 server-ip 1.1.1.1 server-port any**

Related Commands

The following table summarizes other commands related to the **apps ssl** command:

Task	Command
Displays all SSL statistics	**#show apps ssl stats**
Displays a specified SSL private key.	**# show apps ssl key alias key1**
Displays all SSL keys.	**# show apps ssl key all**

Task	Command
Displays a specified SSL service.	**# show apps ssl service alias service1**
Displays all SSL services.	**# show apps ssl service all**
Displays SSL service statistics.	**# show apps ssl service stats**
Displays specified SSL service statistics.	**# show apps ssl service stats alias service1**
Displays all SSL service statistics.	**# show apps ssl service stats all**
Displays statistics associated with the passive SSL decryption GigaSMART group. For descriptions of the session statistics, refer to the "*Flow Ops Report Statistics for Passive SSL Decryption*" topic in the *GigaVUE-FM User's Guide*.	**# show gsgroup flow-ops-report alias gsg1 type ssl-decryption any**
Matches the specified hostname and displays the corresponding session details.	**# show gsgroup flow-ops-report alias gsg1 type ssl-decryption match hostname www.xxxxx.com**
Uploads the flow ops report file to the specified remote server. Specify the remote server path and password to access the server.	**# show gsgroup flow-ops-report alias gsg1 type ssl-decryption any upload scp://username@10.22.0.79:/path/foldername**
Displays GSOP for Passive SSL decryption.	**# show gsop by-application ssl-decrypt**
Displays GSOP statistics for Passive SSL decryption.	**# show gsop stats by-application ssl-decrypt**
Displays all or specified SSL server certificate statistics.	**#show apps ssl server-cert stats all/{alias}**
Deletes a specified SSL private key.	**(config) # no apps ssl key alias key1**
Deletes a comment associated with a specified SSL private key.	**(config) # no apps ssl key alias key1 comment**
Deletes all SSL keys.	**(config) # no apps ssl key all**
Deletes a specified SSL service.	**(config) # no apps ssl service alias service1**
Deletes all SSL services.	**(config) # no apps ssl service all**

apps ssl-profile

Use the **apps ssl-profile** command to configure SSL Profile parameters.

The **apps ssl-profile**command has the following syntax:

```
apps ssl-profile alias <name>
    comment "string"
    mtls <disable; default: "disable">
```

```
cipher <string; default: TLS_AES_128_GCM_SHA256>
version <string; default: TLS1.3>
 exit
```

The following table describes the arguments for the **apps ssl-profile** command:

Argument	Description
alias <alias>	Specifies the SSL proile alias. For example: **(config) # apps ssl-profile alias ssl-profile**
mtls <disable; default: "disable">	MTLS is disabled by default.
cipher <string; default: TLS_AES_128_GCM_SHA256>	Only SHA 256 is supported.
version <string; default: TLS1.3>	Only TLS Version1.3 is supported.

Related Commands

The following table summarizes other commands related to the **apps ssl-profile** command:

Task	Command
Displays a specified SSL profile.	**# show apps ssl-profile [alias <alias>]**
Displays all SSL profiles.	**# show apps ssl-profile [all]**

apps tcp-profile

Use the **apps tcp-profile** command to configure TCP profile parameters.

The **apps tcp-profile**command has the following syntax:

The following table describes the arguments for the **apps tcp-profile** command:

```
apps tcp-profile alias <name>
    comment "string"
    keep-alive-timer <30-7200; default 60>
    selective-ack <enable/disable; default: enable>
   syn-retries <1-6; default 3>
   exit
```

Argument	Description
alias <alias>	Specifies the TCP alias. For example: **(config) # apps tcp-profile alias tcp-profile1**
keep-alive-timer <30-7200; default 60>	Specifies the time duration between keep alive messages. The value ranges from 30-7200. The default value is 60.
selective-ack <enable/disable; default: enable>	Enables or disables selective acknowledgement. The default is **enable**.
syn-retries <1-6; default 3>	Enter the value for number of times the SYN has to be tried. The value ranges from 1 to 6. The default value is 3.

Related Commands

The following table summarizes other commands related to the **apps tcp-profile** command:

Task	Command
Displays a specified TCP profile.	**# show apps tcp-profile [alias <alias>]**
Displays all TCP profiles..	**# show apps tcp-profile [all]**

apps trust store

Use the **apps trust store** command to create, update, or delete a client trust store used to validate client certificates for inbound and hybrid mTLS.

The **apps trust store**command has the following syntax:

apps trust-store alias <trust-store> type client fetch [append | replace]

no apps trust-store alias <trust-store>

The following table describes the arguments for the **apps trust store** command:

Argument	Description
alias <trust-store> type client fetch [append | replace]	Creates or updates a client trust store used to validate mTLS client certificates; fetch downloads the client CA bundle, append adds certificates to the existing store, and replace overwrites the store. For example: **(config) # apps trust-store alias client_trust_1 type client fetch append scp://user@host:/path/client_ca.pem**
no apps trust-store alias <trust-store>	Deletes a client trust store that is no longer required; the alias must not be referenced by any Inline SSL profile. For example: **(config) # no apps trust-store alias client_ trust_1**

Related Commands

The following table summarizes other commands related to the **apps trust store**command:

Task	Command
Displays the certificates in the specified trust store, including subject CN, issuer CN, fingerprint, expiration date, and trust-store type (server or client). Use this command to verify the contents of a client trust store configured for mTLS. For example: **(config) # show apps trust-store alias client_trust_1**	**show apps trust-store alias <trust-store>**
Displays the full certificate details for a single entry in the specified trust store, selected by fingerprint. Use this to inspect individual client CA certificates used for mTLS. For example: **(config) # show apps trust-store alias client_trust_1 certificate fingerprint 7F:41:4A:DD**	**show apps trust-store alias <trust-store> certificate fingerprint <fingerprint>**

banner

Required Command-Line Mode = Configure

Use the **banner** command to add customizable text banners that appear when a user logs into the GigaVUE-OS® HC Series node.

Enclose the banner string in double quotation marks, if the string is longer than one word.

The **banner** command has the following syntax:

```
banner
  login <string> | default>
  login-local <string>
  login-remote <string>
  motd <string> | default>
```

The following table describes the arguments for the **banner** command:

Argument	Description
login <string> | default>	Configures the login banner. This banner is displayed at the login prompt before you log in to the GigaVUE-OS® HC Series node. Depending on the banner, it can provide information about the node you are logging in to. For example: **(config) # banner login "Pacific Call Center"** To reset the login banner to the default: **(config) # banner login default**
login-local <string>	Configures the local login banner. This banner is displayed at the login prompt before you log in locally to the node, such as from the console. For example: **(config) # banner login-local Welcome**
login-remote <string>	Configures the remote login banner. This banner is displayed at the login prompt before you log in remotely to the node, such as through SSH. For example: **(config) # banner login-remote “Welcome users”**
motd <string> | default>	Configures a message of the day banner. This banner is displayed after you have logged in to the node. For example: **(config) # banner motd “April 14 Welcome”** To reset the message of the day banner to the default: **(config) # banner motd default**

Related Commands

The following table summarizes other commands related to the **banner** command:

Task	Command
Displays the currently configured banners.	**# show banner**
Clears the login banner.	**(config) # no banner login**
Clears the message of the day banner.	**(config) # no banner motd**

battery

Use the **battery** command to execute battery test .The battery command has following syntax:

battery
battery test [execute | cancel] [ID]

The following table describes the arguments for the **battery** command

Argument	Description

battery test [execute][ID]	Starts the battery test for the specified battery ID. For example:
	`# battery test execute abcde1234567890`
	You cannot simultaneously start another BET process by giving another battery ID. For example:
	`# battery test execute pqrst1234567890Battery Exercise Test is already in-progress for ID abcde1234567890...`

battery test [cancel] [ID]	Cancels the battery test for the specified battery ID.
	`# battery test cancel abcde1234567890`
	`Battery Exercise Test canceled for abcde1234567890`

Related Commands

The following table summarizes other commands related to the **battery** command:

Task	Command
Displays the battery status	**# show battery status**

battery optimization

Use the **battery optimization** command to optimize battery drain for G-TAP A-SF21 when the power supply is down . Battery optimization can be done in two methods such as:

- Switch-Off Unused Ports
- Switch -Off Monitor Port.

The **battery optimization** command has following syntax:

battery optimization
<policy> <port-group-id | port-group-list > <level>

Argument	Description
<policy>	Defines the policy to be used for optimizing battery which can be : **1.** unused-port --> Help string for this policy is "disable unused port" **2.** monitor-port --> --> Help string for this policy is "disable monitor port"
<port-group-id | port-group-list | global>	The system level configuration to optimize the battery by : • Port group -ID- which is X1,X2,X3,X4. • Port-group-list – a series of port groups can also be defined for battery optimization.
<level>	Defines the battery level at which battery optimization must be started. The battery level values are as follows: • NORMAL • WARNING • ALARM. The default value will be ALARM when you first configure the port groups.

Related Commands

The following table summarizes other commands related to the **battery optimization** command:

Task	Command
Displays the battery optimization	**# show battery optimization [port-group-id | port-group-list]**
Clears the battery optimization	**#no battery optimization**

battery optimization global

Use the **battery optimization global** command to initiate cpu hibernation for G-TAP A Series 2 .

The **battery optimization global** command has following syntax:

#battery optimization global
cpu-hibernate <value> sleep-time <value>

The following table provides argument and description **battery optimization global**

Argument	Description
global	Refers to the system-level configuration
cpu-hibernate	Refers to the battery level after which device goes to hibernation state.The value specifies the battery level percentage which ranges from between 85 to 25. When this value is hit, the

Argument	Description
<value>	device will go into hibernation state.
sleep-time <value>	This indicates the periodic CPU Hibernation sleep duration.The value specifies the sleep time in minutes which ranges from 6 to 30.

Related Commands

The following table summarizes other commands related to the **battery optimization global** command:

Task	Command
Displays the current CPU hibernation configuration setting.	**# show battery optimization global**
Disables cpu hibernation	**#no battery optimization global cpu-hibernate**

bond

Required User Level = Admin

Use the **bond** command to configure bonding interfaces and modes. Bonding is a Linux networking feature. Only basic functions are available in the CLI, such as defining a bonding interface and adding a peer interface to a bonding interface. These functions are useful for cluster management, to provide redundant cluster control links.

The **bond** command has the following syntax:

```
bond <bonding interface>
  down-delay-time <milliseconds>
   link-mon-time <milliseconds>
   mode <balance-rr | backup | balance-xor | balance-xor-layer3+4 | broadcast | link-agg |
      link-agg-layer3+4 | balance-tlb | balance-alb>
   up-delay-time <milliseconds>
```

The following table describes the arguments for the **bond** command:

Argument	Description
bond <bonding interface>	Creates a bonding interface. For example:

Argument	Description
	(config) # bond bond0
down-delay-time <milliseconds>	Configures a down delay time for the bonding interface, in milliseconds. This is the amount of time to wait before disabling a peer interface after a link failure has been detected.
link-mon-time <milliseconds>	Configures a link monitoring frequency for the bonding interface, in milliseconds.
mode <balance-rr | backup | balance-xor | balance-xor-layer3+4 | broadcast | link-agg | link-agg-layer3+4 | balance-tlb | balance-alb>	Specifies the type of mode of the bonding interface. The modes are as follows: • **balance-rr**—round robin load balancing • **backup**—backup fault tolerant mode • **balance-xor**—XOR load balancing • **balance-xor-layer3+4**—XOR load balancing Layer 3 + 4 mode • **broadcast**—broadcast fbondault tolerant mode • **link-agg**—link aggregation mode (IEEE 802.3ad) • **link-agg-layer3+4**—link aggregation Layer 3 + 4 mode • **balance-tlb**—adaptive transmit load balancing • **balance-alb**—adaptive load balancing
up-delay-time <milliseconds>	Configures an up delay time for the bonding interface, in milliseconds. This is the amount of time to wait before enabling a peer interface after a link recovery has been detected.

Related Commands

The following table summarizes other commands related to the **bond** command:

Task	Command
Creates a bonding interface.	**(config) # bond bond0**
Adds a peer interface to a specified bonding interface.	**(config) # interface eth0 bond bond0**
Deletes an interface from a specified bonding interface.	**(config) # no interface eth0 bond bond0**
Displays configuration information about all bonding interfaces.	**# show bonds**
Displays configuration information for a specified bonding interface.	**# show bonds bond0**
Deletes a specified bonding interface.	**(config) # no bond bond0**

boot

Required Command-Line Mode = Enable
Required User Level = Admin

The GigaVUE-OS® HC Series node has two partitions, each with a separate image installed. Use the **boot** command to select which of the system's two images to use at the next boot, what to do if the selected image does not boot correctly, and so on. This improves system stability when installing new software—you can preserve the existing software on one partition and fall back to it if necessary.

The **boot** command has the following syntax:

(missing or bad snippet)

The following table describes the arguments for the **boot** command:

Argument	Description
bootmgr password <password>	Specifies a password for access to the boot manager. The boot manager is available during system startup when connected over the console port. If you enter the boot manager, you can override the default image selected using **boot system**.
next fallback-reboot enable	Enables fallback reboot if the selected image does not load correctly. When this option is enabled, the GigaVUE-OS HC Series node will fall back and boot the other image if the selected image does not load correctly. This setting is enabled by default.
system location <1 | 2> next	Specifies the image to load the next time the system is booted. You can specify either one of the two available partitions by number or use the next argument to select the next image available after the previously booted one. Use **show system** to see the available images and their corresponding partitions. For example, the following command selects the image stored in Partition 1 for the next boot: **(config) # boot system location 1** **Note**: The **boot system** command is the same as the **image boot** command.

Related Commands

The following table summarizes other commands related to the **boot** command:

Task	Command
Displays system information, including boot settings.	**# show system**
Displays information on system images, including the last boot partition, next boot partition, and currently installed system images.	**# show images**

Task	Command
Disables fallback reboot if the configuration file cannot be applied after an upgrade attempt.	**(config) # no boot next fallback-reboot enable**
Resets the next boot location to the currently active one.	**(config) # no boot system next**

card (GigaVUE-OS® HC Series)

Required Command-Line Mode = Configure
Required User Level = Admin

Use the **card** command to manage GigaVUE-OS® HC Series line cards. You can **configure** a card, causing the system to recognize a newly installed card and make it available for use, shut down cards in preparation for removal, and show card status.

NOTE: You must run the **card** command to configure all cards as part of the installation of a new system. You must also use the **card** command to configure a newly installed card. Refer to the **Hardware Installation Guide** for a description of the initial configuration procedure.

The **card** command has the following syntax:

```
card <all [box ID] | slot <slot ID>>
  alarm buffer-threshold <0-100>
  down
  fabric-hash advanced
  filter-template <<filter template alias> | defaults>
  mode <32x | 2q>
  product-code <card product code>
  set buffer alpha <alpha value>
```

NOTE: In a cluster environment, you must enter the **<slot id>** in the **<box ID>/<slot ID>** format. For example, **card 13/8** configures the card in slot 8 on box ID 13.

The following table describes the arguments for the **card** command:

Argument	Description
<all [box ID] | slot <slot ID>>	Configures the card(s) at the specified slot(s) or on the specified node, preparing them for use. When you issue this command, the system reads the card type and hardware information, allowing the card(s) to be recognized by the system. This command is often used during the initial configuration of a system. For example: **(config) # card all**

Argument	Description
	Note: Reset is triggered in the GigaVUE-HC1-Plus platform when performing 'card slot <4>' for the rear GigaSMART card or 'card slot <2>' for the front GigaSMART card or 'card slot <3>' for the front GigaSMART card. Since the eport state is still up, the card will appear up until the reset is triggered and the socket connection is lost. This limitation applies to the front GigaSMART card on 'card slot <2>' in the GigaVUE-HCT platform.
alarm buffer-threshold <0-100>	Sets the alarm buffer threshold for a slot, as a percentage from 1 to 100. The default is 0, which disables the threshold. Buffer usage increases when there is congestion in the chassis. This argument configures the threshold at which an SNMP trap is sent. For example: **(config) # card slot 4/1 alarm buffer-threshold 75** When the threshold reaches 75%, a trap notifies you that the buffer usage has crossed the configured threshold. **Note**: On the GigaVUE-HC3, this command configures the same alarm buffer threshold on all the slots in the chassis.
slot <slot ID> down	Shuts down the specified line card and prepares it for removal. Refer to the line card removal procedure in the **Hardware Installation Guide** for instructions.
slot <slot ID> fabric-hash advanced	Enables advanced fabric hashing on the specified card and slot. For deployment, contact Technical Support. For example: **(config) # card slot 3/5 fabric-hash advanced**
slot <slot ID> filter-template <<filter template alias> | defaults>	Applies a filter template to a specified line card or module. The filter template must have already been created. To create a filter template, refer to filter-template. For example: **(config) # card slot 1/3 filter-template ipv6+mac** The **defaults** keyword is a special alias which switches back to the predefined filter templates. For example: **(config) # card slot 1/3 filter-template defaults** Switching from a user-defined filter template to the predefined filter templates has the following restrictions: • the number of existing rules cannot exceed the rules limitation of the predefined filter templates • the existing rules must match any of the predefined filter template qualifiers
slot <slot ID> mode <32x | 2q>	Sets the operating mode for a PRT-H00-Q02X32 line card. You can only change the mode for the line card if it has been unconfigured using **no card <slot id>**. The available operating modes are as follows: **2q Mode:** • **Two** 40Gb (QSFP+) ports (q1..q2) • **Twenty-four** 10Gb/1Gb (SFP+/SFP) ports (x5..x28) **32x Mode:**

Argument	Description
	• **Thirty-two** 10Gb/1Gb (SFP+/SFP) ports (x1..x32) For example: **(config) # card slot 11/3 mode 32x** Refer to the **Hardware Installation Guide** for details and examples on changing the mode of the PRT-H00-Q02X32 line card. For card modes on GigaVUE TA Series nodes, refer to card (GigaVUE-OS®TA Series).
slot <slot ID> product-code <card product code>	Configures the card with a product code (for offline provisioning). For example: **(config) # card slot 3/5 product-code 132-00DW**
slot <slot ID> set buffer alpha <alpha value>	For GigaVUE-OS-TA1 and GigaVUE-OS-TA10, allows ports to consume a greater percentage of the common shared buffer pool. For deployment, contact Technical Support. **Note**: This command is not supported in a cluster environment.

Related Commands

The following table summarizes other commands related to the **card** command:

Task	Command
Displays the configuration and status of all cards on the local node.	**# show cards**
Displays the card details on a specified box.	**# show cards box-id 1**
Displays the card details on a specified slot.	**# show cards slot 2/1**
Unconfigures all cards in the chassis.	**(config) # no card all**
Unconfigures all cards on a specified box ID.	**(config) # no card all box-id 1**
Unconfigures a specified card at a slot so that it is no longer recognized by the system. You can reenable the card with the **card** command. This is handy when changing the type of card installed in a slot, permanently removing a card, or changing a system's box ID.	**(config) # no card slot 2**
Unconfigures the card alarm buffer threshold.	**(config) # no card slot 2 alarm buffer-threshold**
Reactivates the card.	**(config) # no card slot 2 down**
Disables advanced fabric hashing on the specified card and slot.	**(config) # no card slot 3/5 fabric-hash advanced**
Displays buffer profile current information.	**# show profile current buffer all**

Task	Command
Displays a minute of buffer profile history information.	**# show profile history buffer 2/1/x1 min**
Displays buffer usage by box ID.	**# show buffer box-id 2**
Displays buffer usage by port ID.	**# show buffer port 2/1/x1**
Displays buffer usage by port ID and direction.	**# show buffer port 2/1/x1 rx**
Displays buffer usage by slot.	**# show buffer slot 2/1**

NOTE: Card mismatch occurs when:

- a card that has a particular mode is replaced with another card with a different mode or with a card that does not support a mode
- a configured card is replaced with another card without unconfiguring the initial card. Refer to the *Hardware Installation Guides* for card replacement procedures.

card (GigaVUE-OS® TA Series)

Required Command-Line Mode = Configure
Required User Level = Admin

Use the **card** command to specify how the GigaVUE TA Series node uses its physical 40Gb ports. This command also applies to white box nodes with GigaVUE-OS.

The **card** command has the following syntax for GigaVUE-OS-TA1, GigaVUE-OS-TA10, and white box:

card slot <slot ID> mode <48x | 56x | 64x>

GigaVUE TA Series nodes only have a single card, so the slot value is always set to 1.

The following table summarizes the available options:

Card Mode	GigaVUE-OS on a white box			
	q1	q2	q3	q4

48x (default)	40Gb (q1)	40Gb (q2)	40Gb (q3)	40Gb (q4)
56x	10Gb (x49..x52 on breakout panel)	10Gb (x53..x56 on breakout panel)	40Gb (q3)	40Gb (q4)
64x	10Gb (x49..x52)(x49..x52 on breakout panel)	10Gb (x53..x56)(x53..x56 on breakout panel)	10Gb (x57..x60 on breakout panel)	10Gb (x61..x64 on breakout panel)

Use break-out cables or breakout panels (PNL-M341 or PNL-M343). For breakout panel information, refer to the *GigaVUE TA Series* **Hardware Installation Guide.**

chassis

Required Command-Line Mode = Configure
Required User Level = Admin

Use the **chassis** command to assign the chassis a box ID or other parameters, or migrate an existing configuration to the chassis.

The **chassis** command has the following syntax:

```
chassis
  buffer
   box-id <box ID>
     gdp <enable | disable>
     mgmt-intf discovery <cdp | lldp | all>
          mgmt-intf garp
     mode <normal | 100G [left | right]>
    serial-num <serial number> [gdp <enable | disable> | type <hc1 | hc3 | hc3-v2 | ly2r | itac |
tacx | ta200|ta25|ta25a>]
  mgmt-intf garp
              tag-mode <single | double>
beacon led enable | disable
 migrate box-id <box ID> [serial-num <serial number>]
```

The following table describes the arguments for the **chassis** command:

Argument	Description
buffer	Configures the buffer for the chassis.
box-id <box ID>	Configures the box ID for the chassis. The box ID identifies the node in the system. The allowable range of box-id is from 1 to 64. **Note**: The box ID must be configured before line cards can be configured and ports enabled. Unique box IDs are required for each node in a cluster.

Argument	Description
	For example: **(config) # chassis box-id 2**
gdp <enable | disable>	Enables or disables Gigamon discovery on the chassis. • **enable**—Enables Gigamon discovery (GDP) on the chassis. • **disable**—Disables Gigamon discovery (GDP) on the chassis. The default is disable for chassis.
mgmt-intf discovery <cdp | lldp | all>	Enables discovery protocols on the management interface of a device, as follows: • **cpd**—Enables CDP on the management interface. • **lldp**—Enables LLDP on the management interface. • **all**—Enables both CDP and LLDP on the management interface. For example: **(config) # chassis box-id 1 mgmt-intf discovery cdp** Port discovery is disabled by default. For more information about the discovery protocols, refer to the "*Enable Discovery Protocols*" section in the GigaVUE Fabric Management Guide.
mgmt-intf garp	Enables Gratuitous ARP on the management interface of the specified box ID in a cluster. Gratuitous ARP is disabled by default. Examples: **(config) # chassis box-id 2 mgmt-intf garp** For more information about Gratuitous ARP, refer to the "*Enable Gratuitous ARP on Management Interface*" section in the GigaVUE Fabric Management Guide.
mode <normal | 100G [left | right]>	The mode of the chassis as follows: • **normal**—Specifies the normal mode. This mode is for all platforms, except 100Gb modules and PRT-HC0-CO2. • **100G**—Specifies the 100G mode on both the left and the right columns of the node. This mode is only for 100Gb modules and PRT-HC0-CO2. Optionally, you can configure the following: o **left**—Specifies the 100G mode on the left column only o **right**—Specifies the 100G mode on the right column only The default is **normal**. To change the mode, first remove the existing configuration from the chassis, then configure the mode, as follows: **(config) # no chassis box-id 1** **(config) # chassis box-id 1 mode 100G** or **(config) # no chassis box-id 2** **(config) # chassis box-id 2 serial-num C0D55 type hc3-v2 mode 100G** In **normal** mode, the maximum number of ports on each column is 48.

Argument	Description
serial-num <serial number> [gdp <enable | disable> | type <hc1 | hc1p | hct | hc3 | hc3-v2 ly2r | ta10 | ta10a | ta40 | hc1 | itac | tacx | ta200|ta25|ta25a>]	Configures the serial number of the node and the type of node. Enables Gigamon discovery on the chassis. The gdp configuration is as follows: • **enable**—Enables Gigamon discovery (GDP) on the chassis. • **disable**—Disables Gigamon discovery (GDP) on the chassis. The default is disable for chassis. The node types are as follows: • **hc1—**GigaVUE-HC1 • **hc1p**—GigaVUE-HC1-Plus • **hct**—GigaVUE-HCT • **hc3**—GigaVUE-HC3 • **ly2r**—Certified Traffic Aggregation White Box • **itac**—GigaVUE-TA100 • **ta200**—GigaVUE-TA200 • **ta25**- GigaVUE-TA25 • **ta25a**-GigaVUE-TA25A The serial numbers of the node are displaying in the **show chassis** command. Examples: **(config) # chassis box-id 2 serial-num 1C80-1000** **(config) # chassis box-id 2 serial-num 1C80-1000 type hc3**
tag-mode <single | double>	Configures the VLAN tag mode for the chassis. • **Single**—Enables single-tag mode and provides visibility only to a single VLAN tag and untagged traffic. • **double**—Enables double tag mode and provides visibility to double VLAN tag, single VLAN tag, and untagged traffic. Examples: **(config) # chassis box-id 1 serial-num T0006 type hc1-v2 gdp enable tag-mode double**
beacon led enable | disable	Enables or disables locator led on the chassis. • **enable**—Enables locator led on the chassis. • **disable**—Disables locator led on the chassis. Refer to the System LED's section in the GigaVUE-TA400E Hardware Guide.
migrate box-id <box ID> [serial-num <serial number>]	Moves the configuration in one chassis to a new chassis. Running the **chassis migrate box-id <box ID>** command copies the packet distribution settings. Make sure you save the newly migrated configuration with **write memory** when finished and then use **reload** to reboot the system. The **migrate** argument copies over all settings that use the box ID. This includes, card, chassis, port, map, map rule, port-pair, map-passall, tool-mirror, and GigaStream settings—anything that uses a box ID as part of its setting. Settings that do not use the box ID are also stored on the CC1 and come over automatically. This

Argument	Description
	includes IP settings for the Mgmt port, AAA servers, SNMP configuration, and logging settings. Include the **serial-num** argument to specify the serial number of the node. Examples: **(config) # chassis migrate box-id 2** **(config) # chassis migrate box-id 2 serial-num 80386**

Related Commands

The following table summarizes other commands related to the **chassis** command:

Task	Command
Displays configuration and operational status for the chassis.	**# show chassis**
Displays configuration for all line cards or modules on this box.	**# show chassis box-id 1**
Displays faceplate numbering on a Certified Traffic Aggregation White Box. This command only applies on a white box.	**# show chassis box-id 1 faceplate-numbering**
Displays port discovery information on the management interface for the specified box ID.	**# show chassis box-id 1 mgmt-intf discovery # show chassis box-id 1 mgmt-intf discovery brief**
Displays the Gratuitous ARP configuration information on the management interface of the specified box ID.	**# show chassis box-id 1 mgmt-intf garp**
Displays Gigamon discovery information for all card.	**# show gdp all**
Displays Gigamon discovery information for all cards in a specified box.	**# show gdp box-id 1**
Displays Gigamon discovery information in table format.	**# show gdp brief**
Displays Gigamon discovery neighbors.	**# show gdp neighbor**
Displays Gigamon discovery information for a list of ports.	**# show gdp port-list 1/1/x1**
Displays Gigamon discovery information for a list of ports in table format.	**# show gdp port-list 1/1/x1 brief**
Displays Gigamon discovery for a card at a specified slot.	**# show gdp slot 2**
Displays Gigamon discovery for a card at a specified slot in table format.	**# show gdp slot 2 brief**
Deletes the active configuration on the chassis. You can also use the **no chassis box-id** command to remove the chassis configuration. This is useful when moving a control card from one chassis to another—after installing the card in the new chassis, use **no chassis box-id** to remove the stored configuration. Refer to the **Hardware Installation Guide** for the detailed procedure.	**(config) # no chassis**

Task	Command
Deletes the chassis configuration for the specified box ID. Use this command before changing the chassis mode.	**(config) # no chassis box-id 1**
Disables the port discovery on the management interface of the specified box ID.	**(config) # no chassis box-id 1 mgmt-intf discovery**
Disables the Gratuitous ARP on the management interface of the specified box ID in a cluster.	**(config) # no chassis box-id <boxid> mgmt-intf garp**

clear

Required Command-Line Mode = Configure

Use the **clear** command to clear statistics, caches, counters, or to reset information.

The **clear** command has the following syntax:

```
clear
   aaa authentication attempts <all | user <username>> [no-clear-history | no-unlock]
  apps
    asf stats <alias <alias> | all>
    inline-ssl
      caching <cert-validation | url>
      monitor stats
      session debug vport <vport alias>
      session summary
    netflow
      exporter stats [alias <alias> | all]
      monitor
          cache [alias <alias> | all]
        stats [alias <alias> | all]
    ssl service stats [alias <alias> | all]
  arp
  gsgroup
    flow-ops
      flow-sampling <alias <alias> | all>
      flow-filtering <alias <alias> | all>
      flow-sip <alias <alias> | all>
      ssl-decryption <alias <alias> | all>
    stats [alias <alias> | all]
  gsop stats
    alias <alias>
    all
    by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking
| slicing | strip-header | trailer | tunnel-decap | ssl-decrypt>
    by-gsgroup <GS group alias>
  hb-counters <alias <alias> | all>
  ip
    destination stats all
    interface stats [alias <alias> | all]
  ipv6 neighbors
```

load-balance port-group stats <alias <alias> | all>
map stats <alias <alias> | all>
nhb-counters <alias <alias> | all> pcap all port
** phy port-list <port-list>**
** quadphy <port ID>**
** stats <all | box-id <box ID> | port-list <port list> | slot <slot ID>>**
tunnel <l2gre | vxlan>
tunnel-endpoint stats port-list <GigaSMART group alias>
vport stats [alias <alias> | all]

The following table describes the arguments for the **clear** command:

Argument	Description
aaa authentication attempts <all | user <username>> [no-clear-history | no-unlock]	Clears authentication attempts.
apps asf stats <alias <alias> | all>	Resets ASF statistics.
apps inline-ssl caching <cert-validation | url>	Clears inline SSL persisted records and files. Also clears the GigaSMART shared memory caches.
apps inline-ssl monitor stats	Clears inline SSL monitor mode statistics.
apps inline-ssl session debug vport <vport alias>	Reserved for internal use.
apps inline-ssl session summary	Clears inline SSL session summary statistics.
apps netflow exporter stats [alias <alias> | all] monitor cache [alias <alias> | all] stats [alias <alias> | all]	Resets NetFlow exporter or monitor.
apps ssl service stats [alias <alias> | all]	Resets Passive SSL decryption information.
arp	Clears all dynamic ARP entries in the IPv4 cache.

Argument	Description
gsgroup **flow-ops** **flow-sampling <alias <alias> | all>** **flow-filtering <alias <alias> | all>** **flow-sip <alias <alias> | all>** **ssl-decryption <alias <alias> | all>** **stats [alias <alias> | all]**	Resets gsgroup information. Resets gsgroup flow sampling. Resets gsgroup flow filtering. Resets gsgroup SIP session report statistics. Resets gsgroup SSL decryption. Resets gsgroup statistics.
gsop **alias <alias>** **all by-application [add-header |** **dedup | apf | asf |** **flow-sampling |** **flow-filtering |** **lb | masking |** **slicing | strip-header |** **trailer | tunnel-decap |** **ssl-decrypt]** **by-gsgroup <GS group alias>**	Resets gsop information.
hb-counters <alias <inline tool alias> | all>	Clears heartbeat counters on inline tools.
ip **destination stats**	Resets IP destination statistics.

Argument	Description
all **interface stats <alias <alias> | all>**	Resets IP interface statistics.
ipv6 neighbors	Clears all dynamic entries in the IPv6 neighbors cache.
load-balance port-group stats <alias <port-group name> | all>	Clears load balancing port group statistics.
map stats <alias <alias> [rule <rule ID>] | all>	Clears map counters.
nhb-counters <alias <alias> | all>	Clears negative heartbeat counters on inline tools.
pcap all	Clears all hardware and software entries for packet capture, such as control and data plane entries.
port **phy port-list <port-list>** **quadphy <port ID>** **stats [all | box-id <box ID> |** **port-list <port list> |** **slot <slot ID>]**	Resets or clears ports as follows: • **phy**—Resets the specified ports in the **<port-list>** on the PRT-H00-Q02X32 line card on GigaVUE-OS HD Series nodes. This command resets ports without having to reload the entire line card. • **quadphy**—Resets all ports in the group of four sequential ports (quad) containing the specified port on the PRT-H00-Q02X32 line card on GigaVUE-OS HD Series nodes. The line card has eight quads of four ports each starting with x1..x4, x5..x8, to x29..x32. **Note**: Because the **clear port quadphy** command interrupts traffic on all four ports in the quad, use it only after trying to reset a port using the **clear port phy** command. • **stats**—Clears port statistics reported in the **show port-stats** command for the specified ports. You can specify all ports, a **<port-list>**, or all ports in a specified slot. For example, the following command clears the port stats for all ports in slot 3: **(config) # clear port stats slot 3** You can define the **<port-list>** using any combination of the following standard conventions (refer to Port Lists Definition in the GigaVUE-OS for more information): • **port-id**—<bid/sid/pid> • **port-alias**—<port-alias> • **port-list**—<bid/sid/pid_x..pid_y> (range) | <bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list)
tunnel <l2gre | vxlan>	Reserved for future use.
tunnel-endpoint	Resets tunnel endpoint statistics information.

Argument	Description
stats port-list <GS group alias>	
vport stats [alias <alias> | all]	Resets vport stats information.

cli

Use the **cli** command to configure the behavior of the command-line interface, including how long a session can be inactive before the system logs it out automatically, paging settings, terminal size and so on.

Most of the cli commands can be set for either the current **session** or as the new **default** for all sessions. The command-line mode required for each is different as follows:

- Changing a cli **session** setting requires only Standard command-line mode.
- Changing a cli **default** setting requires Configure command-line mode.

The **cli** command has the following syntax:

```
cli
   clear-history
  default
    auto-logout <number of minutes>
    init-resize
    paging enable
    progress enable
    prompt <confirm-reload | confirm-reset | confirm-unsaved | empty-password>   session
    auto-logout <number of minutes>
    paging enable
    progress enable
    terminal
      length <number of lines>
      resize
      type <ansi | console | dumb | linux | screen | vt52 | vt100 | vt102 | vt220 | xterm>
      width <number of characters>
```

The following table describes the arguments for the **cli** command:

Command	Description
clear-history	Clears the command history for the current user.
default auto-logout <number of minutes>session	Specifies how long a CLI session can remain inactive before it is automatically logged out by the system. Use the corresponding command to configure this either for the current **session** or as the new CLI **default**. For example:

Command	Description
auto-logout <number of minutes>	**(config) # cli session auto-logout 180** Use a value of 0 to specify that sessions never log out automatically due to inactivity. For example, to specify that the current CLI session never expires due to inactivity, use the following command: **(config) # cli session auto-logout 0** **Note**: Commands that produce a lot of output such as **show log continuous** might timeout depending on the auto-logout configuration.
default init-resize	Directly reads terminal dimensions from the device on CLI startup.
default paging enable **session paging enable**	Specifies whether paging is enabled for screen output that exceeds the current window size. When paging is enabled (the default), you can page through output using the same features as the Linux programs less and more. Press the **h** key to see paging options. Figure 2Viewing Paging Options provides an example. Use the corresponding command to configure this either for the current **session** or as the new CLI **default**.
default progress enable **session progress enable**	Specifies whether the CLI should provide progress updates for operations that take a long time to complete. Use the corresponding command to configure this either for the current **session** or as the new CLI **default**.
prompt <confirm-reload | confirm-reset | confirm-unsaved | empty-password>	Configures when the CLI should prompt you for input, as follows: • **confirm-reload**—Prompts for confirmation before rebooting. • **confirm-reset**—Prompts for confirmation before resetting to factory state. • **confirm-unsaved**—Confirms whether to save unsaved changes before rebooting. • **empty-password**—Prompts for a password if none is specified.
session terminal length <number of lines> **session terminal resize** **session terminal type <ansi | console | dumb | linux | screen | vt52 | vt100 | vt102 | vt220 | xterm>** **session terminal width <number of characters>**	Configures the terminal output to specified dimensions, as follows: • **length**—Overrides the autodetected length of the terminal. Specify the **length** in lines. • **resize**—Resets the terminal dimensions to the current window. • **type**—Sets the terminal dimensions to a specified terminal type. • **width**—Overrides the autodetected width of the terminal. Specify the **width** in characters. Refer also to terminal.

Related Commands

The following table summarizes other commands related to the **cli** command:

Task	Command
Displays CLI options.	**# show cli**
Displays CLI command history.	**# show cli history**
Displays CLI command history for a specified number of lines.	**# show cli history 10**
Does not automatically log users out due to keyboard inactivity.	**(config) # no cli default auto-logout**
Does not read terminal dimensions from the device on CLI startup.	**(config) # no cli default init-resize**
Disables paging.	**(config) # no cli default paging enable**
Disables progress updates.	**(config) # no cli default progress enable**
Does not prompt for confirmation before rebooting.	**(config) # no cli default prompt confirm-reload**
Does not prompt for confirmation before resetting to factory state.	**(config) # no cli default prompt confirm-reset**
Does not save unsaved changes before rebooting.	**(config) # no cli default prompt confirm-unsaved**
Assumes there is no password if none is specified in a pseudo-URL for SCP.	**(config) # no cli default prompt empty-password**
Does not automatically log users out due to keyboard inactivity.	**(config) # no cli session auto-logout**
Disables paging.	**(config) # no cli session paging enable**
Disables progress updates.	**(config) # no cli session progress enable**
Clears the terminal type.	**(config) # no cli session terminal type**

clock

Required Command-Line Mode = Configure

Use the **clock** command to set the system's local time, date, and time zone.

The **clock** command has the following syntax:

```
clock
   set <hh:mm:ss> [<yyyy/mm/dd>]
  timezone <zone> [<zone word> [<zone word> [<zone word>] [<zone word>]]clock]
```

The following table describes the arguments for the **clock** command:

Command	Description
set <hh:mm:ss> [<yyyy/mm/dd>	Sets the time and date for the system clock. The time must be specified but the date is optional. If you do not supply a date, it remains as currently set.
timezone <zone> [<zone word> [<zone word> [<zone word>] [<zone word>]]]	Specifies the timezone for the local system clock. You can define the timezone as follows: • **UTC**—Enables the use of UTC, for example: • (config) # clock timezone UTC • **UTC-offset**—Defines the timezone as an offset from UTC with the UTC-offset argument. For example, the following command sets the timezone as eight hours earlier than UTC: • (config) # clock UTC-offset UTC-8 • **Location**—Specifies a particular location's timezone to use. The easiest way to do this is to build the available list of cities by adding a space and a question mark after each subsequent zone word. For example, enter **(config) # clock timezone ?** to see the list of available locations. Select a location and use the same question mark technique to see the next available zone words. This way, you can build out to the exact location you need. For example: • (config) # clock timezone America North United_States Pacific • (config) # clock timezone Asia Eastern Hong_Kong

Related Commands

The following table summarizes other commands related to the

Task	Command
Displays clock settings.	**# show clock**
Resets the timezone to the default (GMT).	**(config) # no clock timezone**

cluster

Required Command-Line Mode = Enable or Configure

Use the **cluster** command to create and manage clusters. A cluster is a group of GigaVUE HC Series nodes operating as a unified fabric with packets entering a port on one node capable of being sent to any destination port on another node.

Refer to the "*Creating and Managing Clusters*" section in the *GigaVUE Fabric Management Guide* for details on setting up all aspects of a cluster.

NOTE: If you rename a cluster using the GigaVUE-OS CLI, the rename does not reflect in the GigaVUE-FM. You cannot rename a cluster from GigaVUE-FM. Refer to the GigaVUE Fabric Management Guide for details.

The easiest way to configure a cluster is with the **config jump-start** script described in the **Hardware Installation Guide**. This script walks you through the configuration of the essential commands required to create a cluster, such as the Cluster ID, Cluster Name, and Cluster Management IP Address (a virtual IP address used to access the leader, no matter which physical node is performing that role at the current time).

The **cluster** command has the following syntax:

```
cluster
  enable
  id <cluster ID>
  interface <interface> <ipv4 | ipv6>
 ip-ver-switch diagnostics
  cluster ip-ver-switch
  leader
    address
      primary ip <cluster leader IP> [port <leader port number>]
      secondary ip <cluster leader IP> [port <leader port number>]
      vip <cluster leader vip> <netmask | mask length>
    auto-discovery
    connect timeout <seconds>
    interface <interface>
    preference <1-100>
    yield
  name <cluster name>
  port <cluster port number>
  reload [box-id <box ID>] | [force] | [node-id <node ID>]
  reload sequential
  remove <node ID>
  shared-secret <shared secret>
  shutdown
  startup-time <cluster startup time (secs)>
```

The following table describes the arguments for the **cluster** command:

Argument	Description
enable	Enables cluster support for the node as follows: • If the currently specified cluster ID does not match an existing cluster, creates a new cluster with this node becoming the leader. • If the currently specified cluster ID matches an existing cluster, the node joins the existing cluster. For example: **(config) # cluster enable** To disable cluster support for the node, meaning that the node will leave the cluster, use the following: **(config) # no cluster enable**
id <cluster ID>	Specifies the cluster ID for the node. When joining an existing cluster, configure the cluster ID for the node to match the existing cluster's ID. The cluster ID can contain up to 32 alphanumeric characters and can include the hyphen (-) special character. For example: **(config) # cluster id 100**
interface <interface> <IPv4 | IPv6>	Specifies the interface for the cluster. The interface can be **eth0** (the Management port), **eth1**, (the dedicated cluster Management port on GigaVUE-HC1), and GigaVUE-HC3or **inband**. For example: **(config) # cluster interface eth1** **Note**: All nodes in a GigaVUE HC Series cluster must use the same interface. Only the **eth0** interface is supported for Layer 3 out-of-band manual discovery. Specify the interface as either IPv4 or IPv6 on top of which the devices communicate with each other. Default value is IPv4. If the interface is not specified, it will be considered as IPv4.
ip-ver-switch diagnostics	Establishes a server-client relationship between the leader and the member nodes of the cluster and checks if all devices in the cluster are reachable and if there are any firewall or routing issues.on
cluster ip-ver-switch	Changes the communication protocol version from IPv4 to IPv6 and vice-versa
leader address primary ip <cluster leader IP> [port <leader port	Sets options relating to the leader in the cluster. The leader role on the GigaVUE HC Series is not statically assigned to a single node. Instead, another node in the cluster can take on the leader role if the situation requires it (for example, if both the leader and the current standby nodes go down). When a new node becomes the leader, it takes ownership of the virtual IP address used for leader access to the cluster. Use the leader argument to set the following options: • **address primary ip**—Specifies the IP address used by the leader in the cluster to allow nodes on a different subnet to manually discover the cluster leader. This is the address

Argument	Description
number>] **secondary ip <cluster leader IP> [port <leader port number>]** **vip <cluster leader vip>** **<netmask |** **mask length>** **auto-discovery** **connect timeout <seconds>** **interface <interface>** **preference <1-100>** **yield**	used to join the cluster. For example: **(config) # cluster leader address primary ip 192.168.1.52 port 60102** • **address secondary ip**—Specifies the IP address used by the standby node in the cluster to allow nodes on a different subnet to manually discover the standby or the potential leader of the cluster. For example: **(config) # cluster leader address secondary ip 192.168.1.54 port 60102** • **address vip**—Specifies the virtual IP address and netmask or mask length used by the node in the cluster performing the leader role. This is the address you use to access the cluster. Only IPv4 address is supported for the VIP. Note that IPv4 is used for communication between the nodes in a cluster. Examples: **(config) # cluster leader address vip 192.168.1.25 /24** • **auto-discovery**—Enables auto-discovery of the cluster leader. By default, auto-discovery is enabled. For example: **(config) # cluster leader auto-discovery** To allow nodes on a different subnet to manually discover the cluster, set auto-discovery to no. For example: **(config) # no cluster leader auto-discovery** • **connect timeout**—Specifies the time available for a node residing on a different subnet to discover a new leader to allow nodes on a different subnet to manually discover the cluster. When a leader fails and the standby is promoted to the new leader, the node is allowed to discover the new leader within the time specified in the timeout value. The default is 15 seconds. The values range from 10 to 120 seconds. For example: **(config) # cluster leader connect timeout 30** This parameter applies to nodes on a different subnet to allow them to join a cluster. • **interface**—Specifies the **ethx** interface to be used for cluster management traffic for the virtual IP. The valid values are eth0 and eth1. **Note**: Clustering is not supported on eth2 interfaces. For example: **(config) # cluster leader interface eth1**
leader preference <1-100> yield (continued)	• **preference**—Specifies how likely a node is to claim the leader role during the leader contention process (for example, across a cluster reload). Higher values are more likely to claim the leader role; lower values are less likely. The cluster leader preference can be configured to a preference value between 1 and 100. Set higher preference values for nodes with more processing power. Use settings from 10 to 100 for leader, standby, and normal roles. Use preference settings from 1 to 9 for normal nodes that are excluded from taking the leader or standby role.

Argument	Description
	Starting in software version 4.5, the preference cannot be set to 0. A node with a preference of 0 in an earlier software version will be changed to 1 after an upgrade to 4.5 or higher. For example: **(config) # cluster leader preference 80** • **yield**—Yields the current leader role to the node performing the standby role. If you are not sure which node is currently performing the standby role, use **show cluster global brief** to see the list of all the nodes in the cluster, including their current role. For example: **(config) # cluster leader yield**
name <cluster name>	Specifies the cluster name. This is the cluster-level equivalent of a hostname. It must match for all nodes in a cluster. The cluster name can contain up to 64 alphanumeric characters and can include the hyphen (-) special character. For example: **(config) # cluster name cluster-100**
port <port number>	Specifies the service port number used for the cluster. The port specified must match for all nodes in the cluster. The range of numeric values for the port is from 1025 to 65535. For example: **(config) # cluster port 60102**
reload **box-id <box ID>** **force** **node-id <node ID>**	Reloads/reboots either the entire cluster or a specified node in the cluster, as follows: • Reboot the entire cluster with **cluster reload**. • Reload a specified node by specifying either its box ID or its node ID. You can see a list of these values for all nodes in the cluster with the **show cluster global brief** command. • Use the **force** argument to force an immediate reboot. For example: **(config) # cluster reload box-id 14**
reload sequential	Reloads all the nodes in a cluster in a sequential order. This command may take additional time to reload the nodes as compared to the **cluster reload** command because when you run the **cluster reload sequential** command, the leader waits for each of the nodes in the cluster to reload and join the cluster again. **Note**: It is recommended that you use this command, instead of the **cluster reload** command, when you want to reload the nodes in an Inband cluster. For example: **(config) # cluster reload sequential**
remove <node ID>	Removes the specified node from the cluster using the node ID. The **remove** argument can only be used when logged in to the leader, either directly or through the VIP address. For example:

Argument	Description
	(config) # cluster remove 20
shared-secret <shared secret>	Specifies the shared secret used for message authentication between all nodes in the cluster. The secret must match across all nodes. The shared secret can be from 16 to 64 alphanumeric characters and can include special characters, such as !, @, #, $, %, ^, &, *, (,), _, and +. The default value is the following string: • 1234567890123456 For example: **(config) # cluster shared-secret MyShared1234567890**
shutdown	Puts all nodes in the cluster in a down state (similar to **reload halt**). The **shutdown** argument can only be used when logged in to the leader, either directly or through the VIP address. For example: **(config) # cluster shutdown**
startup-time <cluster startup time (secs)>	Specifies the maximum number of seconds allowed for cluster startup. The range of numeric values for the startup time is from 0 to 2147483647 seconds. The default is 180 seconds. For example: **(config) # cluster startup-time 360**

Related Commands

The following table summarizes other commands related to the **cluster** command:

Task	Command
Displays cluster information for a specified box.	**# show cluster box-id 1**
Displays global cluster configuration state.	**# show cluster configured**
Displays global cluster run state.	**# show cluster global**
Displays global cluster run state in table format. Use this CLI command on the leader, standby, or normal node to display the maximum (Max) and Used cost units across a cluster	**# show cluster global brief**
Displays cluster history log.	**# show cluster history**
Displays cluster history log for a specified box.	**# show cluster history box-id 1**
Displays local cluster run state.	**# show cluster local**
Displays error status of local node.	**# show cluster local error-status**
Displays run state information about the leader.	**# show cluster leader**

Task	Command
Displays information about a node.	**# show cluster node 1**
Displays run state information about the standby node.	**# show cluster standby**
Leaves the cluster.	**(config) # no cluster enable**
Resets cluster ID to the default.	**(config) # no cluster id**
Resets interface to the default for cluster service.	**(config) # no cluster interface**
Resets the cluster leader primary IP address to the default.	**(config) # no cluster leader address primary ip**
Resets the cluster leader secondary IP address to the default.	**(config) # no cluster leader address secondary ip**
Resets the cluster leader virtual IP address (VIP) to the default.	**(config) # no cluster leader address vip**
Disables cluster leader auto-discovery.	**(config) # no cluster leader auto-discovery**
Resets cluster leader interface to the default.	**(config) # no cluster leader interface**
Resets the cluster name to the default.	**(config) # no cluster name**
Resets the cluster service port to the default.	**(config) # no cluster port**
Does not authenticate messages.	**(config) # no cluster shared-secret**
Resets cluster startup time to the default.	**(config) # no cluster startup-time**

configuration

Required Command-Line Mode = Configure

Use the **configuration** command to manage configuration files on the GigaVUE-OS® HC Series node—separate arguments let you perform a wide variety of related tasks, including:

- Save, copy, and delete configuration files.
- Upload and retrieve configuration files from external hosts using FTP, TFTP, or SCP.
- Display the contents of a configuration file.
- Load a named configuration file.
- Return to a previous configuration file's settings.

Configuration File Types

There are two types of configuration files on the GigaVUE-OS® HC Series node—**standard** configuration files and **text** configuration files (known as **command files**):

- **Standard** configuration files can be used to store and apply a set of settings with the **configuration switch-to** command.
- **Text** configuration files are not really configuration files at all—instead, they are lists of CLI commands used to build a particular configuration. Text configuration files are useful both for both troubleshooting and backup purposes—you can quickly see the commands that built a particular configuration, or you can store regular backups of text files containing the commands on an external host. Text configuration files can also be applied in the CLI using the **configuration text file <filename> apply** command.

  You work with text configuration files using the **configuration text** command and its arguments.

Information Excluded from Text Configuration Files

For security reasons, text configuration files do not include plaintext passwords, such as SMTP passwords, AAA keys (RADIUS or TACACS+), private keys in RSA/DSA identities. Because of this, they cannot completely restore a given configuration using **configuration text file <filename> apply.**

Reserved Empty Database File

The empty database file, empty_db_file_dnu, is a reserved file. Do not use (dnu) this filename in any database operation such as **configuration write to** or **configuration switch-to** commands as the filename is removed when the node is reloaded.

The **configuration** command has the following syntax:

```
configuration
  audit max-changes <number>
  copy <source filename | initial> <destination filename>
  delete <filename | initial>

  fetch <download URL> <filename>   jump-start
  move <source filename | initial> <destination filename>
  new <filename> [factory [keep-basic] [keep-connect]]
  revert saved
 switch-to <filename | initial>
  text
    fetch <download URL>
      apply [discard] [fail-continue] [filename <filename>] [overwrite] [verbose]
      filename <filename> [apply] [fail-continue] [overwrite] [verbose]
      overwrite [apply] [fail-continue] [filename <filename>] [verbose]
    file <filename>
       apply [fail-continue] [verbose]
      delete
      rename <filename>
```

```
    upload <upload URL>
  generate
    active running <only-traffic> <save <filename>> | <upload <upload URL>>
    active saved <only-traffic> <save <filename>> | <upload <upload URL>>
    file <filename | initial> <save <filename>> | <upload <upload URL>>
upload <initial | active> <upload URL>   write [local | to <filename>] [no-switch]
```

The following table describes the arguments for the **configuration** command:

Argument	Description
audit max-changes <number>	Sets the maximum number of configuration changes that will be logged for the audit feature.
copy <source filename | initial> <destination filename>	Makes a copy of the specified configuration file. Specify filenames for both the source and destination filenames. For example, the following command copies the configuration file named **gigavue** to a new file named **mybackup**: **(config) # configuration copy gigavue mybackup** Note that you cannot copy over the active configuration file. However, you can copy it to a new file—if you do so, the original remains active.
delete <filename | initial>	Deletes the named configuration file. You cannot delete the active configuration file. **Tip:** Type a space and question mark after the **delete** argument to see the list of configuration files available for deletion. For example: **(config) # configuration del ?**
fetch <download URL> <filename>	Retrieves a saved configuration file from a remote host. Use HTTP(S), FTP, TFTP, or SCP to retrieve the file. The format for the download URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command retrieves the configuration file named myconfig from the FTP server at 192.168.1.10 using the **robh** account with the **xray** password: **(config) # configuration fetch ftp://robh:xray@192.168.1.10/myconfig** You can also use the **<filename>** argument to give the retrieved file a new name on the GigaVUE HC Series node. For example, the following command retrieves **myconfig** and names it **newconfig** on the node: **(config) # configuration fetch scp://bbochy:catch1@192.168.1.75/myconfig newconfig** **Note**: You cannot retrieve a file with the same name as the currently active configuration file.
jump-start	Runs the configuration wizard for the initial setup of GigaVUE-OS nodes. Refer to the **Hardware Installation Guide** for details. The **configuration jump-start** automatically starts and forces a password change. The system administrator must change the password on the default **admin** account.
move <source	Renames the specified configuration file. For example, the following command

Argument	Description
filename | initial> <destination filename>	renames myconfig as newconfig: **(config) # configuration move myconfig newconfig** **Note**: The currently active configuration file cannot be either the source or destination of a **move**.
new <filename> [factory [keep-basic] [keep-connect]]	Creates a new configuration file with the specified filename. The parameters specify what configuration, if any, is carried forward from the current running configuration, as follows: • **keep-basic**—Preserves licenses, SSH host keys, and CMC rendezvous configuration. • **keep-connect**—Preserves anything necessary to maintain network connectivity to the system, such as interfaces, routes, and ARP. You can select one or both or neither after **factory**. If no optional parameters are specified, the default is **keep-basic**.
revert saved	Reverts the system configuration to a previously saved state, either from the last saved configuration file or from the factory settings. Use the **saved** argument to revert the running configuration to the settings in the last saved configuration file. For example: **(config) # configuration revert saved**
switch-to <filename | initial>	Loads the named configuration file, making it the active file. For example, the following command loads the myconfig configuration file: **(config) # configuration switch-to myconfig**
text **fetch <download URL>** **apply [discard] [fail-continue]** **[filename <filename>]** **[overwrite]** **[verbose]** **filename** **<filename> [apply]** **[fail-continue]**	Retrieves a saved text configuration file from a remote host. Use HTTP(S), FTP, TFTP, or SCP to retrieve the file. The format for the download URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command retrieves the text configuration file named **textconfig** from the FTP server at 192.168.1.40 using the **sven** account with the **svenpass** password: **(config) # configuration text fetch ftp://sven:svenpass@192.168.1.40/textconfig** The arguments are as follows: • **filename**—Gives the retrieved file a new name on the GigaVUE HC Series node. For example, the following command retrieves **textconfig** and names it **newtextconfig** on the node: **(config) # configuration text fetch scp://bbochy:catch1@192.168.1.75/myconfig filename newconfig** Use the following arguments to control how errors are handled and reported: o **fail-continue**—Include this option if you want to continue executing commands if one fails. If this option is not included, the process halts on the first error. o **verbose**—Include this option if you want to see all commands printed to the screen as they are applied. If this option is not included, only commands resulting in errors (or output) appear in the CLI.

Argument	Description
[overwrite] [verbose] overwrite [apply] [fail-continue] [filename <filename>] [verbose]	○ **overwrite**—Overwrites the filename if the destination filename already exists. ○ **apply**—Retrieves the text configuration file and applies it to the running configuration at the same time. For example, here is the previous command with the **apply** switch added: **(config) # configuration text fetch scp://bbochy:catch1@192.168.1.75/myconfig filename newconfig apply verbose fail-continue** If you include the **apply** argument, you can also include the **discard** option to specify that the text file be discarded once the commands are applied. Note that the **discard** option is mutually exclusive with the **filename** option. For example, here is a version of the previous command that will apply the text configuration file and discard it afterwards: **(config) # configuration text fetch scp://bbochy:catch1@192.168.1.75/myconfig apply discard verbose fail-continue**
text file <filename> apply [fail-continue] [verbose] delete rename <filename> upload <upload URL>	Applies a text-based configuration file to the running configuration, or deletes, renames, or uploads a specified file to an external server using FTP, TFTP, or SCP. **Applying a Text Configuration File** For example, the following command applies the text configuration file named textconfig: **(config) # configuration text file textconfig apply** Use the following arguments to control how errors are handled and reported: • **fail-continue**—Include this option if you want to continue executing commands if one fails. If this option is not included, the process halts on the first error. • **verbose**—Include this option if you want to see all commands printed to the screen as they are applied. If this option is not included, only commands resulting in errors (or output) appear in the CLI. **Deleting a Text Configuration File** Use the **delete** argument to delete a text configuration file. For example: **(config) # configuration text file myconfig delete** **Renaming a Text Configuration File** Use the **rename** argument to rename a text configuration file. For example: **(config) # configuration text file myconfig rename your config** **Uploading a Text Configuration File** Use the **upload** argument to send a text configuration file to an external server using FTP, TFTP, or SCP. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command uses SCP to upload **myconfig** to 192.168.1.212: **(config) # configuration text file myconfig upload scp://bposey:catch1@192.168.1.212**
text generate	Generates a text-based configuration file from the system configuration. The text file can be saved locally or uploaded to a remote host.

Argument	Description
active running <only-traffic> <save <filename>> | <upload <upload URL>> **active saved <only-traffic> <save <filename>> | <upload <upload URL>>** **file <filename | initial> <save <filename>> | <upload <upload URL>>**	Text configuration files list the CLI commands used to create a particular configuration. They can be applied using the **configuration text file <filename> apply** command. The available sources for a text configuration file are as follows: • **active running**—The currently running configuration, complete with any unsaved changes to the active configuration file. • **active saved**—The last saved version of the active configuration file. • **active running only-traffic**—The only-traffic text configuration of the currently running configuration, complete with any unsaved changes to the active configuration file. Only-traffic means the text file will contain only the traffic-related configuration. • **active saved only-traffic**—The only-traffic text configuration of the last saved version of the active configuration file. Only-traffic means the text file will contain only the traffic-related configuration. • **file <filename>**—The named configuration file of an inactive saved configuration. Each text configuration file can be saved or uploaded as follows: • **save <filename>**—Specifies saving the text file to persistent storage. • **upload <upload URL>**—Specifies uploading the text file to a remote host. Use FTP, TFTP, SCP, or SFTP to upload the file. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command uploads a text configuration file based on the active running configuration and uploads it to an FTP server at 192.168.1.49: **(config) # configuration text generate active running upload ftp://myuser:mypass@192.168.1.49** Uploaded text configuration files are automatically named with a timestamp in epoch format. For example: **config-text-1308003659** You can also generate text configuration files and save them to local storage. For example: **(config) # configuration text generate active running save myfile.txt** You can also generate traffic-only text configuration files and save them locally. For example: **(config) # configuration text generate active running only-traffic save myfile.txt** **(config) # configuration text generate active saved only-traffic save myfile.txt** You can also generate traffic-only text configuration files and upload them. For example: **(config) # configuration text generate active running only-traffic upload scp://username@192.168.1.105** **(config) # configuration text generate active saved only-traffic upload ftp://myuser:mypass@192.168.1.49**

Argument	Description
upload <inital> | active <upload URL>	Uploads a configuration file to a remote host. Use FTP, TFTP, or SCP to upload the file. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command sends the configuration file named **gigavue** to the FTP server at 10.160.10.212 using the **ramrod** account with the **xyz123** password: **(config) # configuration upload gigavue ftp://ramrod:xyz123@10.160.10.212** You can also use the **active** argument to upload the active configuration file. The uploaded file will have the same name as the active file. **(config) # configuration upload active scp://bposey:catch1@192.168.1.212**
write [local | to <filename>] [no-switch]	Saves the running configuration to storage. You can save locally or to a currently active file or a named file. For example: • **(config) # configuration write** Saves the running configuration to the active configuration file. If you run this command on a leader node, it is propagated to all the nodes in the cluster. • **(config) # configuration write local** On a system with clustering, saves the running configuration on the local node. • **(config) # configuration write to myconfig** Saves the running configuration to a named file and makes it active. In the above example, **myconfig** is the name of the file. This is local to the node on which the command is run and is not propagated to other nodes in the cluster. • **(config) # configuration write to myconfig no-switch** Saves the running configuration to the **myconfig** file and leaves the current configuration file active.

Related Commands

The following table summarizes other commands related to the **configuration** command:

Task	Command
Displays commands to recreate active saved configuration.	**# show configuration**
Displays settings for configuration change auditing.	**# show configuration audit**
Displays a list of configuration files.	**# show configuration files**
Displays the commands in a configuration file to recreate the configuration.	**# show configuration files file1**
Does not exclude commands that set default values.	**# show configuration full**
Displays commands to recreate current running configuration.	**# show configuration running**

Task	Command
Does not exclude commands that set default values.	**# show configuration running full**
Displays names of available text-based configuration files.	**# show configuration text files**
Displays the commands necessary to recreate the current running configuration.	Use either of the following: • **(config) # write terminal** • **# show running-config**

configure

Required Command-Line Mode = Enable

Use the **configure** command to enter Configure mode. Refer to Command Line Modes for more information.

The **configure** command has the following syntax:

configure terminal

The following table describes the arguments for the **configure** command:

Argument	Description
terminal	Enters configuration mode. For example: **co t(config) #**

Related Commands

The following table summarizes other commands related to the **configure** command:

Task	Command
Exits configuration mode.	**(config) # exit** or **(config) #** no configure

coreboot

Required Command-Line Mode = Configure

Use the **coreboot** command to install the core boot binary on a GigaVUE-OS-TA100, GigaVUE-OS-TA100-CXP, GigaVUE-HC1, or GigaVUE-HC3 node. The Basic Input/Output System (BIOS) image can be manually upgraded from the CLI using this command.

NOTE: This command only applies to GigaVUE-OS-TA100, GigaVUE-OS-TA100-CXP, GigaVUE-HC1, and GigaVUE-HC3 nodes.

The GigaVUE-OS-TA100 is introduced in software version 4.6.01. The GigaVUE-HC1 is introduced in software version 4.8.00. The GigaVUE-OS-TA100-CXP is introduced in software version 4.8.01. The GigaVUE-HC3 is introduced in software version 5.0.00.

The BIOS image is packaged with the software image. Use the **coreboot** command to upgrade the BIOS, when required for enhancements.

The **coreboot** command has the following syntax:

coreboot install

The following table describes the arguments for the **coreboot** command:

Argument	Description
install	Installs core boot binary from the active/booted image. For example: **(config) # coreboot install**

Related Commands

The following table summarizes other commands related to the **coreboot** command:

Task	Command
Displays the BIOS image from which the system booted.	**(config) # show version**

crypto

The GigaVUE-OS node by default generates and uses a self-signed certificate to provide HTTPS access for GigaVUE-FM to communicate and manage GigaVUE-OS node. It also facilitates the user to utilize ACME to manage web certificates. Use the **crypto** command to configure and manage certificates for the GigaVUE-OS® HC Series node's built-in Web server, performing the following tasks:

- Generate the certificate and key pairs on the GigaVUE HC Series node. This overwrites the existing certificate and key pair regardless of whether the previous certificate and key pair was self-signed or user added. You can specify how long the new self-signed certificate lasts with the **days-valid** argument.
- Replace a signed certificate with one created by an administrator or generated by a 3rd party certificate authority.
- Generate a certificate request and upload it to a specified URL. Default values for the certificate request can be configured.
- The user can also utilize ACME to issues, renew and revoke web certificates. Also, you can set the ACME certificate parameters.

NOTE: X509 Certificates signed with SHA1 are not supported.

The **crypto** command has the following syntax:

```
crypto
acme client clear
cert-req-msg
    generate upload <upload URL>
    generation default
       country-code <country code>
      days-valid <number of days>
      email-addr <email address>
      key-size-bits <number of bits>
      locality <locality name>
      org-unit <organizational unit name>
      organization <organization name>
      state-or-prov <state or province name>
  certificate
        acme issue box-id <box-id>
            domain <xyz.gigamon.com>
            ca-url <url> |algorithm <rsa-2048 | rsa-4096 | ec-prime256v1 |ec-secp384r1>|
            |renew-days <1-365>| |root-cert <certificate_name>|
     acme params csr
       country-code <country code>
       locality <locality name>
       org-unit <organizational unit name>
       organization <organization name>
       state-or-prov <state or province name>
     acme params timeout <timeout>
      acme renew box-id <box-id> domain <xyz.gigamon.com>
      acme revoke box-id <box-id>domain <xyz.gigamon.com>
       ca-list default-ca-list name <CA list name> [system-self-signed]
  default-cert name <cert name> [system-self-signed]
  generation default
    country-code <country code>
    days-valid <number of days>
    email-addr <email address>
```

```
        key-size-bits <number of bits>
        locality <locality name>
        org-unit <organizational unit name>
        organization <organization name>
        state-or-prov <state or province name>
      name <cert name>
        comment <new comment>
        generate self-signed
          comment <comment>
          common-name <issuer and subject common name>
          country-code <country code>
          days-valid <number of days>
          email-addr <email address>
          key-size-bits <number of bits>
          locality <locality name>
          org-unit <organizational unit name>
          organization <organization name>
          serial-num <serial number>
          state-or-prov <state or province name>
        private-key pem <PEM string>
  private-key pem fetch <url>
        prompt-private-key
        public-cert <comment <comment string>> <pem <PEM string>>
        regenerate [days-valid <number of days>]
        rename <new name>
      system-self-signed regenerate [days-valid <number of days 1-7300>]
```

The following table describes the arguments for the **crypto** command:

Argument	Description
acme client clear	Utilize this command to clear the ACME issued certificate from the system. Once the ACME certificate is deleted, the web server will use the default certificate. This command also cancels the auto-renewal timers that are started by the ACME client in the device.
upload <upload URL>	Generates a certificate request message and uploads the request to the specified URL. The supported formats for upload are: SCP, SFTP, and FTP. For example: **(config) # crypto cert-req-msg generate upload scp://gigatest@192.168.1.2/tmp/Password (if required): ********Successfully uploaded certificate signing request with name 'cert-req-filebWdanb.csr'Successfully uploaded private key with name 'cert-req-filebWdanb.key'**
country-code <country code> **days-valid <number of days>**	Configures default values for certificate request message generation as follows:

Argument	Description
email-addr <email address> key-size-bits <number of bits> locality <locality name> org-unit <organizational unit name> organization <organization name> state-or-prov <state or province name>	• **country-code**—Specifies the default value for country code, in two alphanumeric characters. • **days-valid**—Specifies the default value for days valid. The range is from 1 to 65535 days. • **email-addr**—Specifies the default value for the organization's contact email address, in a string. • **key-size-bits**—Specifies the default value for private key size, in bits, in multiples of 1024. • **locality**—Specifies the default value for locality, in a string. • **org-unit**—Specifies the default value for the organizational unit name, in a string. • **organization**—Specifies the default value for the organization's name, in a string. • **state-or-prov**—Specifies the default value for the state or province, in a string.
certificate acme issue box-id <box-id> domain <xyz.gigamon.com> ca-url <url> |algorithm <rsa-2048 | rsa-4096 | ec-prime256v1 |ec-secp384r1>| |renew-days <1-365>| |root-cert <certificate_name>|	Utilize this command to generate a certificate and its corresponding private key for acme client by configuring the values as follows: • **box-id <box-id>** -Specifies the box-id for which the certificate needs to be issued. • **domain <xyz.gigamon.com>** -Specify the domain name, which will become the subject name as well as the alternate subject name in the certificate. • **ca-url <url>**-The Certificate Authority URL. You can use the following optional values as well: • **|renew-days <1-365>|**-Specifies the number of days (1-365 days) before a certificate expires to initiate the certificate renewal. By default this is 1/3rd days before certificate expiry. • **root-cert <certificate name>**-Configure the root certificate of the CA server.
acme params csr country-code <country code> locality <locality name> org-unit <organizational unit name> organization <organization name> state-or-prov <state or province name>	Configures default values for ACME certificate parameters as follows: • **country-code**—Specifies the default value for country code, in two alphanumeric characters. • **locality**—Specifies the default value for locality, in a string. • **org-unit**—Specifies the default value for the organizational unit name, in a string. • **organization**—Specifies the default value for the organization's name, in a string. • **state-or-prov**—Specifies the default value for the state or province, in a string. For example:

Argument	Description
	(config) # crypto certificate acme params csr country-code IN
acme params timeout <timeout>	This command sets the timeout value (in seconds) for downloading the certificate from the ACME server. The default timeout value is 120 seconds. For example: **(config) crypto certificate acme params timeout 40**
acme renew box-id <box-id> domain <xyz.gigamon.com>	Utilize this command to manually renew an already issued certificate as follows: • **box-id <box-id>** -Specifies the box-id for which the certificate needs to be renewed. • **domain <xyz.gigamon.com>** -Specify the domain name, which will become the subject name as well as the alternate subject name in the certificate.
acme revoke box-id <box-id>domain <xyz.gigamon.com>	Utilize this command to manually revoke an already issued certificate as follows: • **box-id <box-id>** -Specifies the box-id for which the certificate needs to be renewed. • **domain <xyz.gigamon.com>** -Specify the domain name, which will become the subject name as well as the alternate subject name in the certificate.
certificate ca-list default-ca-list name <CA list name> [system-self-signed]	Adds the specified CA certificate to the default CA certificate list.
certificate default-cert name <cert name> [system-self-signed]	Specifies the named certificate as the default certificate for authentication on this node.

Argument	Description
certificate generation default **country-code <country code>** **days-valid <number of days>** **email-addr <email address>** **key-size-bits <number of bits>** **locality <locality name>** **org-unit <organizational unit name>** **organization <organization name>** **state-or-prov <state or province name>**	Configures default values for certificate generation as follows: • **country-code**—Specifies the default value for country code, in two alphanumeric characters. • **days-valid**—Specifies the default value for days valid. The range is from 1 to 65535 days. • **email-addr**—Specifies the default value for the organization's contact email address, in a string. • **key-size-bits**—Specifies the default value for private key size, in bits, in multiples of 1024. • **locality**—Specifies the default value for locality, in a string. • **org-unit**—Specifies the default value for the organizational unit name, in a string. • **organization**—Specifies the default value for the organization's name, in a string. • **state-or-prov**—Specifies the default value for the state or province, in a string.
certificate name <cert name> **comment <new comment>** **generate self-signed** **comment <comment>** **common-name <common name>** **country-code <country code>** **days-valid <number of days>** **email-addr <email address>** **key-size-bits <number of bits>** **locality <locality name>** **org-unit <organizational unit name>** **organization <organization name>** **serial-num <serial number>** **state-or-prov <state or province name>** **private-key pem <PEM string>** **private-key pem fetch <url>** **prompt-private-key** **public-cert <comment <comment string>>** **<pem <PEM string>>** **regenerate [days-valid <number of days>]** **rename <new name>**	Configures options for a named certificate to import into the certificate database as follows: • **cert-name**—Specifies a unique identifier for the certificate. • **comment**—Specifies a comment for an existing certificate. • **generate self-signed**—Generates a named self-signed certificate, as follows: o **comment**—Specifies a comment for the certificate. o **common-name**—Specifies a common name for the certificate, in a string o **country-code**—Specifies the country code, in two alphanumeric characters. o **days-valid**—Specifies the days valid. The range is from 1 to 65535 days. o **email-addr**—Specifies the organization's contact email address, in a string. o **key-size-bits**—Specifies the private key size, in bits, in multiples of 1024. o **locality**—Specifies the locality, in a string. o **org-unit**—Specifies the organizational unit name, in a string. o **organization**—Specifies the organization's name, in a string. o **serial-number**—Specifies the serial number, in a lower-case hexidecimal serial number prefixed with 0x. o **state-or-prov**—Specifies the state or province, in a string. • **private-key**—Adds an RSA private key to a previously

Argument	Description
	imported certificate. • **prompt-private-key**—Prompts for a PEM-encoded string. • **public-cert**—Specifies an alternate certificate, such as one issued by a trusted public signing authority. • **pem <PEM string>**—Specifies a certificate data string in Privacy Enhanced Mail (PEM) format. • fetch <url>—Specifies the remote private key location. • **regenerate**—Regenerates a specified certificate. • **rename**—Renames an existing certificate. **NOTE:** Enclose the contents of the PEM file in quotation marks.
certificate system-self-signed regenerate [days-valid <number of days 1-7300>]	Regenerates a certificate. Certificates are configured to expire after a specified number of days. You can regenerate a certificate with this command, using the **days-valid** argument to specify how long it will be valid before it needs to be regenerated again.

Related Commands

The following table summarizes other commands related to the **crypto** command:

Task	Command
Displays the ACME Certificate information and the recent ACME operation that was performed.	**# show crypto acme client info**
Displays the ACME Certificate parameters and timeout value in seconds.	**# show crypto acme cert-params**
Displays cryptographic configuration and state for all certificates in the certificate database.	**# show crypto certificate**
Displays the list of configured trusted certificates of authority (CA).	**# show crypto certificate ca-list**
Displays the list of supplemental certificates configured for the default system CA certificate.	**# show crypto certificate ca-list default-ca-list**
Displays the currently configured default certificate.	**# show crypto certificate default-cert**
Displays details of the currently configured default certificate.	**# show crypto certificate default-cert detail**
Displays the uninterpreted PEM contents of the currently configured default certificate.	**# show crypto certificate default-cert public-pem**

Task	Command
Displays details of all certificates in the certificate database.	**# show crypto certificate detail**
Displays a specified named certificate.	**# show crypto certificate name mycert**
Displays the uninterpreted PEM contents of all certificates in the certificate database.	**# show crypto certificate public-pem**
Deletes a certificate from the CA certificate trust pool.	**(config) # no crypto certificate ca-list default-ca-list name mycert1**
Reverts to the system-self-signed certificate as the default.	**(config) # no crypto certificate default-cert name system-self-signed**
Deletes a specified certificate.	**(config) # no crypto certificate name system-self-signed**
Deletes the comment on a specified certificate.	**(config) # no crypto certificate name system-self-signed comment**

debug

Required Command-Line Mode = Configure

Use the debug command to generate a system dump file for use with Gigamon Technical Support staff. You can generate the system dump files for all the nodes that are part of a cluster by logging into the leader. However, to download the system dump file for a node, you must log into the respective node.

The **debug** command has the following syntax:

```
debug generate dump
    box-id <box-id>
```

In response, the system will generate an encrypted sysdump file and show you the filename for the leader in a cluster. For example:

```
Generated dump sysdump-newHD-20150105-215155.tgz.gpg
```

NOTE: To view the sysdump files generated for the member nodes in a cluster, run the **# show file debug-dump box-id <box-id>** command.

Related Commands

The following table summarizes other commands related to the **debug** command:

Task	Command
Generates an encrypted debug dump (sysdump) file.	**(config) # debug generate dump**
Generates a debug dump (sysdump) file for the specified box ID that is part of the cluster.	**(config) # debug generate dump box-id <box-id>**
Displays the list of dump files.	**# show file debug-dump**
Displays the list of dump files generated for the specified box ID that is part of the cluster.	**# show file debug-dump box-id <box-id>**
Uploads, emails, or deletes a debug dump file. **Note**: If you email the system dump file, it is automatically sent to all email destinations configured to receive informational events (**info**) with the **email** command (**email notify recipient <email address> class info**).	**(config) # file debug-dump [delete <filename>] [email <filename>] [upload <upload URL>]** The format for the upload URL is as follows: **[protocol]://username [:password]@hostname/path/filename** Use FTP, TFTP, SCP, or SFTP for the upload.
Deletes the specified debug dump file for the mentioned box ID in a cluster.	**(config) # file debug-dump box-id <box-id> delete <filename>**
Deletes all the debug dump files for the specified box ID in a cluster.	**(config) # file debug-dump box-id <box-id> delete-all**
Downloads the debug dump file for the box ID that you have logged in to.	**(config) # get debug-dump**

disable

Required Command-Line Mode = Enable

Use the **disable** command to change from Enable mode to Standard mode. Refer to Command Line Modes for more information.

email

Required Command-Line Mode = Configure

Use the **email** command to configure automatic email notifications for events on the GigaVUE-OS® HC Series node. Separate arguments let you add the mail server/port to use, the account from which emails will be sent, email recipients, and so on. Refer to the **Hardware Installation Guide** for a description of how to configure email notification essentials

The **email** command has the following syntax:

```
email
  auth
    enable
```

```
    password [password]
    username <username>
  autosupport
     enable
    event <event name>
    ssl
      ca-list <none | default-ca-list>
      cert-verify
      mode <none | tls | tls-none>
  dead-letter
    cleanup max-age cleanup <duration>
    enable
  domain <hostname or IP address>
  mailhub <hostname, IPv4, or IPv6 address>
  mailhub-port <port number>
  notify
    event <<event name> | all>
    recipient <email address>
      class <failure | info>
     detail
  return-addr <username>
  return-host
  send-test
  ssl
    ca-list <none | default-ca-list>
    cert-verify
    mode <none | tls | tls-none>
```

The following table describes the arguments for the **email** command:

Command	Description
auth **enable** **password [password]** **username <username>**	Enables the sending of notification emails and specifies the account to be used for authentication with the SMTP server specified by the **email mailhub** command as follows: • **enable**—Enables the sending of notification emails. • **password**—Specifies the password used for authentication with the SMTP server. You can leave the password blank to have the system prompt for a password. • **username**—Specifies the account to be used for authentication with the SMTP server.
autosupport **enable** **event <event name>** **ssl** **ca-list <none | default-ca-list>** **cert-verify** **mode <none | tls |**	Configures auto support, as follows: • **enable**—Enables or disables the sending of emails to the auto support address (by default, Gigamon's Technical Support Department) when failures specified with **email autosupport event <event name>** take place. • **event**—Specifies the events that will trigger an email to the auto support destination. Use **email autosupport event ?** to see the list of available events. • **ssl**—Configures security options for auto support email as follows: o **ca-list**—Configures supplemental CA certificates for verification of server certificates. o **cert-verify**—Verifies server certificates. o **mode**—Configures the type of security to use for autosupport email.

Command	Description
tls-none>	
dead-letter cleanup max-age cleanup <duration> enable	Configures the handling of email notifications that could not be sent (for example, because the mail hub was not configured correctly), as follows: • **cleanup**—Specifies how long to save undeliverable emails. Dead letter files older than the configured maximum are automatically deleted. Specify using **d**(day), **h**(hours), **m** (minutes), and **s**(seconds) values. You can use these either together or by themselves—for example, both **5d6h3m1s** and **5d** are accepted. • **enable**—Enables the saving of undeliverable emails.
domain <hostname or IP address>	Specifies the domain from which notification emails will appear to come from. This name is used together with the **hostname** setting to form the name of the domain included in notification emails. The name is formed using the following rules: • If an email domain is specified, it is used. If the **hostname** has any dots in it, everything to the right of the first dot is removed and the email domain is added. • If an email domain is not specified and the hostname has dots in it, it is used as-is. • If an email domain is not specified and the hostname does not have dots in it, the currently-active system domain name is used.
mailhub <hostname, IPv4, or IPv6 address>	Specifies the address or hostname of the server to use for sending notification emails.
email mailhub-port <port number>	Specifies the TCP port used by the server added using **email mailhub**.
notify event <<event name> | all> recipient <email address> class <failure | info> detail	Specifies the events that will trigger notification emails and where they will be sent as follows: • **event**—Specifies the events that will trigger notification emails. Type **email notify event ?** to see the list of available events. You can also use the **event all** argument to generate notification emails for all available events. Be careful with the **all** argument—the amount of emails generated may be more than is tolerable. • **recipient**—Specifies to whom notification emails should be sent, as well as the types of emails they should receive. Each of the events configured with **email notify event** is classified as either a informational or a failure. Enable the types of emails you want this user to receive with the **class** argument. For example, the following command sends emails to bposey@argus.com for all failures: **(config) # email notify recipient bposey@argus.com class failure** **Note**: No email is sent when **failure** is configured and an **info** event is generated. Use the **recipient detail** argument to specify whether summarized or detailed output should be included in the email. Note that not all events have both summary and detail formats.
return-addr	Specifies the address from with notification emails are sent. If you include the **@** character,

Command	Description
<username>	the address is used as-is (for example, **email return-addr support@mycompany.com**). If the **@** character is not included, the system adds the **<return-host>.<domain>**. The default settings is **do-not-reply**.
return-host	Specifies whether or not to include the hostname in the return address for emails. This only takes effect if the **return addr** entry does not contain an **@** character.
send-test	Sends a test mail to all recipients configured for event and failure notifications with the **email notify** command. This command is useful once you have finished configured email settings and want to make sure they are functioning properly.
ssl ca-list <none | default-ca-list> cert-verify mode <none | tls | tls-none>	Configures the use of SSL for notification emails sent by the GigaVUE HC Series node as follows: • **ca-list**—Configures supplemental CA certificates for verification of server certificates. You can either use only the built-in list (**none**) or supplement it with the default CA list configured using the crypto command (**default-ca-list**). • **cert-verify**—Enables certificate verification for emails sent from the GigaVUE HC Series node. With this option, emails will not be sent if TLS cannot be verified. • **mode**—Specifies the type of security to be used for emails sent from the GigaVUE HC Seriesnode. You can enable plain text emails (**none**), TLS security only (**tls**), or TLS first with plain text as a fallback (**tls-none**). Both TLS options use the default server port (**email mailhub-port**).

Related Commands

The following table summarizes other commands related to the **email** command:

Task	Command
Displays email and notification settings.	**show email**
Displays the events that will trigger notification emails.	**show email events**
Disables authentication for sending email.	**(config) # no email auth enable**
Clears password for SMTP authentication.	**(config) # no email auth password**
Clears username for SMTP authentication (effectively disables authentication until the username set again).	**(config) # no email auth username**
Does not send automatic support notifications through email.	**(config) # no email autosupport enable**
Negates certain email event notification settings.	**(config) # no email autosupport event processcrash** **(config) # no email autosupport event livenessfailure**
Delete supplemental CA certificate list.	**(config) # no email autosupport ssl ca-**

Task	Command
	list
Does not verify server certificates.	**(config) # no email autosupport ssl cert-verify**
Resets autosupport email security mode to the default.	**(config) # no email autosupport ssl mode**
Does not clean up old dead letters based on age.	**(config) # no email dead-letter cleanup max-age**
Does not save dead letter for undeliverable emails.	**(config) # no email dead-letter enable**
Clears email domain override.	**(config) # no email domain**
Clears the configured mail hub.	**(config) # no email mailhub**
Clears the configured mail hub port.	**(config) # no email mailhub-port**
Negates certain email event notification settings.	**(config) # no email notify event all** **(config) # no email notify event firmwarechange** **(config) # no email notify event rxtxerror**
Does not send any notifications to this recipient.	**(config) # no email notify recipient friend@gmail.com**
Does not send certain types of events to this recipient.	**(config) # no email notify recipient friend@gmail.com class failure** **(config) # no email notify recipient friend@gmail.com class info**
Sends summarized event emails to this recipient.	**(config) # no email notify recipient friend@gmail.com detail**
Resets the return address to the default.	**(config) # no email return-addr**
Does not include hostname in return address for email notifications.	**(config) # no email return-host**
Deletes supplemental CA certificate list.	**(config) # no email ssl ca-list**
Does not verify server certificates.	**(config) # no email ssl cert-verify**
Resets email security mode to the default.	**(config) # no email ssl mode**

enable

Required Command-Line Mode = Standard

Use the **enable** command to enter Enable mode. Refer to Command Line Modes for more information.

exit

Required Command-Line Mode = Configure

Use the **exit** command to leave Configure mode and return to Enable mode. Refer to Command Line Modes for more information.

fabric advanced-hash

Required Command-Line Mode = Configure

Use the **fabric advanced-hash** command to select the criteria for advanced-hashing behavior on stack GigaStreams and GigaSMART groups. You can configure fabric-advanced hash at the chassis level.

The **fabric advanced-hash** command is supported on the following platforms:

- GigaVUE-HC1
- GigaVUE-HCT
- GigaVUE-HC1-Plus
- GigaVUE-HC3-v2
- GigaVUE-TA25
- GigaVUE-TA25E
- GigaVUE-TA200
- GigaVUE-TA200E
- GigaVUE-TA400
- GigaVUE-TA400E

The **fabric advanced-hash** command has the following syntax:

```
fabric advanced-hash
   all box-id
   default
  fields
      ethertype
      gtpteid
      ip6dst
      ip6nextHeader
      ip6src
      ipdst
      ipsrc
      macdst
      macsrc
      mpls
```

port6dst
port6src
portdst
portsrc
protocol
ingressport
none

The following table describes the arguments for the **fabric advanced-hash** command:

Argument	Description
box-id	Identifies the chassis to which the advanced-hash algorithm will apply. **(config) # fabric advanced-hash box-id 12**
all	Enables all hash criteria fields, including Layer 2, Layer 3, and Layer 4 fields. **Note**: When both Layer 3 (IPv4 or IPv6) and Layer 2 (MAC) fields are enabled for a GigaStream, the hashing behavior depends on the type of incoming traffic. For incoming Layer 3 packets where both Layer 3 (IPv4 or IPv6) fields and Layer 2 (MAC) fields have incremental values, Layer 3 takes precedence. In this case, hashing is performed based on the IPv4 or IPv6 fields, and the MAC fields are not used for hashing. However, for incoming Layer 2 packets that only have incremental MAC addresses, hashing is performed based solely on the MAC fields. **Note**: When both Layer 3 (IPv4) and Layer 2 (MAC) fields are enabled for a GigaStream, the hashing behavior varies based on the packet type. For incoming Layer 3 IPv4 packets with incremental IPv4 and MAC fields, Layer 3 takes precedence, and hashing is performed based on the IPv4 fields, not the MAC fields. For incoming Layer 3 IPv6 packets with incremental IPv6 and MAC fields, hashing is performed based on the MAC fields. In the case of incoming Layer 2 packets with incremental MAC fields, hashing is also based on the MAC fields. **Note**: When both Layer 3 (IPv6) and Layer 2 (MAC) fields are enabled for a GigaStream, the hashing behavior is determined by the type of incoming traffic. For incoming Layer 3 IPv6 packets with incremental IPv6 and MAC fields, Layer 3 takes precedence, and hashing is performed based on the IPv6 fields rather than the MAC fields. For incoming Layer 3 IPv4 packets with incremental IPv4 and MAC fields, hashing is performed based on the MAC fields. Similarly, for incoming Layer 2 packets with incremental MAC fields, hashing is also based on the MAC fields. **(config) # fabric advanced-hash all**
default	Sets the advanced-hash algorithm to its default settings. By default, the advanced-hash algorithm includes source/destination IPv4/IPv6 addresses and ports (**ipsrc**, **ipdst**, **ip6src**, **ip6dst**, **protocol**). For example: **(config) # fabric advanced-hash default**
fields	**(config) # fabric advanced-hash fields <fields>** Specifies the hash criteria. Includes the following options:

Argument	Description
	• **ethertype**—Adds L2 ethertype field. • **gtpteid**—Adds GTP tunnel endpoint identifier. • **ip6dst**—Adds IPv6 destination IP. • **ip6nextHeader**—Adds IPv6 next header field. • **ip6src**—Adds IPv6 source IP. • **ipdst**—Adds IPv4 destination IP. • **ipsrc**—Adds IPv4 source IP. • **macdst**—Adds L2 destination MAC. • **macsrc**—Adds L2 source MAC. • **mpls**—Adds MPLS label (up to three). • **port6dst**—Adds IPv6 destination port. • **port6src**—Adds IPv6 source port. • **portdst**—Adds IPv4 destination port. • **portsrc**—Adds IPv4 source port. • **protocol**—Adds IPv4 protocol. • **ingressport**—Adds ingress port. **Note**: When only Layer 2 (MAC) fields are enabled for a GigaStream, hashing is performed based on the MAC fields, regardless of the packet type. For incoming Layer 3 and above packets that contain incremental IPv4 or IPv6 fields along with incremental MAC fields, the hashing is still based solely on the MAC fields. **ipsrc**, **ipdst**, **ip6src**, and **ip6dst** fields inside an MPLS tunnel can also be used for hashing across GigaStream ports. **GTP TEID** can also be used for hashing across GigaStream ports. **ingress port** can also be used for hashing across GigaStream ports.
none	Clears all fields from the advanced hash. For example: **(config) # fabric advanced-hash box-id 12 none**

Related Commands

The following table summarizes the command related to the **fabric advanced-hash** command:

Task	Command
Displays fabric advanced hash fields for stack GigaStream or gsports	**show fabric advanced-hash**
Displays fabric advanced hash fields for all the chassis in a table format	**show fabric advanced-hash brief**
Displays fabric advanced hash fields for a specified box ID	**show fabric advanced-hash box-id 24**

Task	Command
Displays fabric advanced hash fields for a specified box ID in a table format	**show fabric advanced-hash box-id 24 brief**

NOTE: If the chassis is part of a cluster, then the following show commands will display the fabric advanced hash fields for all the boxes in the cluster:

show fabric advanced-hash
show fabric advanced-hash brief

file

Required Command-Line Mode = Configure

Use the **file** command to manage the debug dump files produced by the **debug** command, packet capture files, and the PCAP files produced by the **tcpdump** command.

Use the **show file debug-dump** command to display a list of available debug dump files. Note that debug dump files may be deleted automatically if disk usage is low.

The **file** command has the following syntax:

```
file
  debug-dump
    delete <filename>
    email <filename>
    upload <filename> <upload URL>
  pcap
    delete <filename>
    delete-all
    upload <filename> <upload URL>    tcpdump
    delete <filename>
    upload <filename> <upload URL>
```

The following table describes the arguments for the **file** command:

Argument	Description
debug-dump delete <filename> email <filename> upload <filename> <upload URL>	Deletes, emails, or uploads a sysdump file produced by the **debug** command as follows: • **delete**—Deletes a file. Type **debug-dump delete ?** to see a list of files available for deletion. • **email**—Sends a sysdump file to all email destinations configured to receive informational events (info) with the **email** command (**email notify recipient <email address> class info**). • **upload**—Sends a debug dump file to a remote host using FTP, TFTP, SCP, or SFTP. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename**
pcap delete <filename> delete-all upload <filename> <upload URL>	Deletes or uploads a packet capture file produced by the **pcap** command as follows: • **delete**—Deletes a specified PCAP file. • **delete-all**—Deletes all PCAP files. • **upload**—Sends a PCAP file to a remote host using FTP, SCP, or SFTP. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example: (config) # show files pcappcap_p1_2018_05_08_17_17.pcap (config) # file pcap upload pcap_p1_2018_05_08_17_28.pcap scp://myNode@10.115.0.100/tftpboot/myName/.Password (if required): ********** To upload the complete PCAP file without errors, delete the packet capture filter. For example: (config) # no pcap alias p1 Refer to pcap.
tcpdump delete <filename> upload <filename> <upload URL>	Deletes or uploads a capture file produced by the **tcpdump** command as follows: • **delete**—Deletes a file. Type **tcpdump delete ?** to see a list of files available for deletion. • **upload**—Sends a tcpdump file to a remote host using FTP, SCP, or SFTP. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename**

Related Commands

The following table summarizes other commands related to the **file** command:

Task	Command
Displays the debug dump files stored on the node.	**show files debug-dump**
Displays PCAP files.	**show files pcap**

Task	Command
Displays filesystem information.	**show files system**
Displays detailed filesystem information.	**show files system detail**
Displays the tcpdump files stored on the nodes.	**show files tcpdump**

filter-template

Required Command-Line Mode = Configure

Use the **filter-template** command to configure filter templates on GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3, GigaVUE-HCT, GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, GigaVUE-TA200, GigaVUE-TA200E, and GigaVUE-OS-TA400.

For more information on filter templates, refer to the *"Flexible Filter Templates"* section in the *GigaVUE Fabric Management Guide*.

The **filter-template** command has the following syntax:

filter-template alias <alias>
comment <comment>
qualifiers <add | remove> <ethertype | inner-ethertype | inner-ip6dst | inner-ip6src | inner-ipfrag | inner-ipver | inner-ipdst | inner-ipsrc | inner-portdst | inner-portsrc | inner-protocol | inner-uda1 | inner-uda2| innervlan | ip6dst | ip6src | ipdst | ipsrc | macdst | macsrc | mpls-label pos {any|1..7|1,2,4..6,7} | mpls-label-id pos {any|1..7|1,2,4..6,7} | mpls-label-exp pos {any|1..7|1,2,4..6,7} | mpls-label-bos pos {any|1..7|1,2,4..6,7} | mpls-label-ttl pos {any|1..7|1,2,4..6,7} | portdst | portsrc | protocol | qset1 | uda1 | uda2 | vlan>

The following table describes the arguments for the **filter-template** command:

Argument	Description
alias <alias>	Supplies an alias for the filter template. The maximum number of characters supported in an alias is 128. Each alias should be unique across the configured filter templates. An alias can be created only after a filter template is defined with one or more qualifiers. An alias cannot contain the forward-slash (/) character.
comment <comment>	Specifies a unique text string that describes the filter template. Comments can be up to 256 characters. Comments must be enclosed in double quotation marks. Comments can be added only after a filter

Argument	Description
	template is defined by specifying an alias and adding one or more qualifiers. For example: **(config) # filter-template alias ft1 comment "UDA filter"**
qualifiers <add | remove> <ethertype | innervlan | ip6dst | ip6src | ipdst | ipsrc | macdst | macsrc | portdst | portsrc | protocol | qset1 | uda1 | uda2 | vlan>	Specifies a list of qualifiers for the filter template. The qualifiers are predefined. The list of qualifiers can consist of one or more qualifiers separated by a space. Specify each qualifier only once. Duplicate qualifiers are not allowed in the list. Examples: **(config) # filter-template alias ft1 qualifiers add uda1 uda2** **(config) # filter-template alias ipv6+mac qualifiers add ip6src ip6dst macsrc macdst** **(config) # filter-template alias ft2 qualifiers add innervlan vlan** **NOTE:** The **uda2** qualifier cannot be specified without the **uda1** qualifier. However, the **uda1** qualifier can be specified with or without the **uda2** qualifier. After a filter template is created, it can be applied to a module. Refer to card (GigaVUE-OS® HC Series). When a filter template is not applied to a line card or module, it can be modified. Qualifiers can be added or removed. For example, to add qualifiers to an existing filter template: **(config) # filter-template alias ft1 qualifiers add vlan** For example, to remove qualifiers from an existing filter template: **(config) # filter-template alias ft1 qualifiers remove uda2**
qualifiers <add | remove> <inner-ethertype | inner-ip6dst | inner-ip6src | inner-ipfrag | inner-ipver | inner-ipdst | inner-ipsrc | inner-portdst | inner-portsrc | inner-protocol | inner-uda1 | inner-uda2 | mpls-label pos {any|1..7|1,2,4..6,7} | mpls-label-id pos {any|1..7|1,2,4..6,7} |	Specifies a list of inner header qualifiers and MPLS header qualifiers for the filter template. The inner header qualifiers and MPLS header qualifiers are predefined and are applicable only to GigaVUE-TA400 devices.

Argument	Description
mpls-label-exp pos {any|1..7|1,2,4..6,7} | mpls-label-bos pos {any|1..7|1,2,4..6,7} | mpls-label-ttl pos {any|1..7|1,2,4..6,7} >	The list of inner qualifiers can consist of one or more inner qualifiers separated by a space. The position chosen as part of mpls-label implicitly applies to to all four MPLS attributes: mpls-label-id, mpls-label-ttl, mpls-label-exp, and mpls-label-bos. If a position is selected for mpls-label, the user must not specify the same position in any of the four MPLS attributes (mpls-label-id, mpls-label-ttl, mpls-label-exp, mpls-label-bos). A filter-template can include a combination of outer and inner qualifiers. Specify each inner qualifier only once. Duplicate inner qualifiers are not allowed in the list. Examples: **(config) # filter-template alias ft1 qualifiers add inner-uda1 inner-uda2** **(config) # filter-template alias ipv6+mac qualifiers add inner-ip6src inner-ip6dst** **(config) # filter-template alias ft2 qualifiers add inner-protocol** **(config) # filter-template alias fft2 qualifiers add ipdst mpls-label-id pos any mpls-label-bos pos 1..2,4..7 mpls-label-exp pos 1,2,4,6..7 mpls-label-ttl pos 2,7 macsrc mpls-label pos 5,3** **(config) # filter-template alias fft2 qualifiers add inner-ipsrc ipsrc mpls-label-exp pos 1,2,4,6..7** After a filter template is created, it can be applied to a module. When a filter template is not applied to a line card or module, it can be modified. Inner qualifiers can be added or removed. For example, to add inner qualifiers to an existing filter template: **(config) # filter-template alias ft1 qualifiers add inner-uda1** For example, to remove inner qualifiers from an existing filter template: **(config) # filter-template alias ft1 qualifiers remove inner-uda2**

Related Commands

The following table summarizes other commands related to the **filter-template** command:

Task	Command
Displays all filter templates in table format.	**show filter-template**
Displays a specified filter template.	**show filter-template alias filt1**
Displays a specified filter template in brief format.	**show filter-template alias filt1 brief**
Displays all filter templates, including default and user-defined templates.	**show filter-template all**
Displays limits of all filter templates in table format.	**show filter-template limit**
Displays limits of all filter templates for GigaVUE-OS nodes. A zero (0) means not supported.	**show filter-template limit all**
Displays limits of the filter templates on the specified box ID.	**show filter-template limit box 1**
Displays limits of the filter templates on the specified slot ID.	**show filter-template limit slot 2**
Displays filter resource usage.	**show filter-resource**
Displays filter resource usage in detail.	**show filter-resource all**
Displays filter resource usage in detail for a specified box ID.	**show filter-resource box-id 1**
Displays filter resource usage in detail for a specified slot ID.	**show filter-resource slot-id 3**
Displays filter resource usage in detail for a specified slot ID on a GigaVUE-OS-TA100.	**show filter-resource slot-id 1/1PS1**
Displays all app filter resource usage in detail.	**show app-filter-rsc**
Displays all app filter resource usage in detail.	**show app-filter-rsc all**
Displays app filter resource usage in detail for a specified box ID.	**show app-filter-rsc box-id 3**
Displays all app filter resource usage in brief format.	**show app-filter-rsc brief**
Displays app filter resource usage in detail for a specified slot ID.	**show app-filter-rsc slot-id 4**
Displays app filter resource usage in detail for a specified slot ID on a GigaVUE-OS-TA100.	**show app-filter-rsc slot-id 1/1PS1**
Displays pseudo-slot port mapping on GigaVUE-OS-TA100.	**show pseudo-slot portmap**

Task	Command
Displays pseudo-slot port mapping on GigaVUE-OS-TA100 for a specified box ID.	**show pseudo-slot portmap box-id 3**
Deletes a specified unused filter template. If the filter template has been applied to a module, it will not be deleted and an error message will be displayed.	**(config) # no filter-template alias filt1**
Deletes all unused filter templates. If any filter templates have been applied to a module, they will not be deleted and an error message will be displayed.	**(config) # no filter-template all**

gigasmart

Required Command-Line Mode = Configure

Use the **gigasmart** command to configure a stack port interface to provide Internet connectivity for a GigaSMART card or module. Internet connectivity is needed for SSL Decryption for inline tools for the URL categorization database. For URL categorization, an IP address must be configured to query the Webroot service.

Internet connectivity is also needed for SSL Decryption for inline tools for a Certificate Revocation List (CRL) to obtain a list of certificates that have been revoked and for an Online Certificate Status Protocol to obtain certificate revocation status.

Finally, Internet connectivity is needed for Hardware Security Module (HSM).

You can configure separate DNS servers for internal and external networks. This ensures better security and privacy management. Use the **app split-dns** command to configure internal DNS servers for each GigaSMART engine port. For details, refer to apps split-dns

NOTE: Only IPv4 addresses are supported.

The **gigasmart** command has the following syntax:

```
gigasmart engine <port-id> proxy-server-profile attach <<Proxy server Alias> interface
    [eth2] | <eth3> [vlan <VLAN ID>]
    <IP address> <netmask> gateway <gateway IP> dns <DNS IP>split-dns <alias>[mtu <1280-
9400>]
     dhcp
 ping
  ping <IP address | hostname> <start | stop>
```

The following table describes the arguments for the **gigasmart** command:

Argument	Description
engine <port-list>	Specifies the GigaSMART engine port on which to configure a stack port interface to provide Internet connectivity for a GigaSMART card or module for SSL Decryption for inline tools. **NOTE:** Only one GigaSMART engine port can be specified in the port list. You can configure a stack port interface on each GigaSMART engine port. For example: **(config) # gigasmart engine 1/1/e1**
proxy-server-profile **attach <alias>** **detach <alias>**	Specifies the proxy server profile that is being attached or detached from the GigaSMART engine.
interface [eth2] | <eth3> [vlan <VLAN ID>]<IP address> <netmask> gateway <gateway IP> dns <DNS IP> split-dns <alias> [mtu <1280-9400>] | dhcp | ping>	Specifies the stack port interface as follows: • **eth2, eth3**—Specifies the stack port interface. The default is eth2. • **vlan**—Specifies an optional VLAN identifier, for Internet connectivity with VLAN. The range of VLAN IDs is from 20 to 4094. • **IP address**—Specifies a static IP address. Only IPv4 addresses are supported. • **netmask**—Specifies the netmask or mask length. For example: 255.255.255.248 or /29. • **gateway**—Specifies the gateway IP address. • **dns**—Specifies the Domain Name Service (DNS). • **split-dns**—Enables the split-DNS profile on the GigaSMART engine port. **NOTE:** Before enabling the split-DNS profile, ensure that you create the profile using the **apps split-dns** command. Refer to apps split-dns. • **mtu**—Specifies the Maximum Transmission Unit (MTU). The range is 1280–9400. The default is 1500. It is recommended that you set the MTU value to 9400 on all platforms to avoid fragmentation. • **dhcp**—Specifies the Dynamic Host Configuration Protocol (DHCP). • **ping**—Specifies to ping using the stack port interface. **NOTE:** Each GigaSMART engine is configured separately. For example, to configure the default (eth2) stack port

Argument	Description
	interface for Internet connectivity: **(config) # gigasmart engine 1/1/e1 interface 1.1.1.2 /24 gateway 1.1.1.1 dns 1.1.1.1 mtu 1500** For example, to configure the eth3 stack port interface on GigaVUE-HC3 for Internet connectivity with VLAN: **(config) # gigasmart engine 1/1/e1 interface eth3 vlan 200 1.1.1.2 /24 gateway 1.1.1.1 dns 1.1.1.1** For example, to configure the default stack port interface using DHCP: **(config) # gigasmart engine 1/1/e1 interface dhcp** For example, to configure the eth3 stack port interface on GigaVUE-HC3 using DHCP: **(config) # gigasmart engine 1/1/e1 interface eth3 dhcp**
ping <IP address | hostname> <start | stop>	Pings using the stack port interface as follows: • **IP address**—Specifies the IP address of the stack port interface. • **start**—Starts the ping request. • **stop**—Stops the ping request and displays the results. **NOTES:** • Always issue the **start** command before the **stop** command. • If you issue the **start** command a second time, it erases the output of the first **start** command. • If you issue the **stop** command a second time, usage help is displayed. • You will not be able to ping the stack port interface from a laptop or client machine because incoming connections to the stack port interface are blocked for security reasons. • In a cluster, ping only works for the leader. You will not be able to ping remote nodes such as standby or normal nodes from the leader. For example: **(config) # gigasmart engine 1/1/e1 ping 1.1.1.2 start** **use command with option "stop" to see ping result** **(config) # gigasmart engine 1/1/e1 ping 1.1.1.2 stop** **PING google.com (216.58.194.174) from**

Argument	Description
	10.115.126.37 mgmt0.12: 56(84) bytes of data. **64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=1 ttl=53 time=3.97 ms** **64 bytes from sfo07s13-in-f174.1e100.net (216.58.194.174): icmp_seq=2 ttl=53 time=3.78 ms** **64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=3 ttl=53 time=3.18 ms** **64 bytes from sfo07s13-in-f174.1e100.net (216.58.194.174): icmp_seq=4 ttl=53 time=3.15 ms** **64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=5 ttl=53 time=3.18 ms** **--- google.com ping statistics ---** **5 packets transmitted, 5 received, 0% packet loss, time 4002ms** **rtt min/avg/max/mdev = 3.156/3.456/3.976/0.357 ms** For example with VLAN: **(config) # gigasmart engine 1/4/e1 interface eth2 vlan 100 ping 1.1.1.1 start** **use command with option "stop" to see ping result** **(config) # gigasmart engine 1/4/e1 interface eth2 vlan 100 ping 1.1.1.1 stop** **PING 1.1.1.1 (1.1.1.1) from 1.1.1.10 mgmt00.100: 56 (84) bytes of data.** **64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=0.065 ms** **64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=0.080 ms** **64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=0.074 ms** **64 bytes from 1.1.1.1: icmp_seq=4 ttl=64 time=0.075 ms** **64 bytes from 1.1.1.1: icmp_seq=5 ttl=64 time=0.091 ms** **--- 1.1.1.1 ping statistics ---** **5 packets transmitted, 5 received, 0% packet loss, time 3996ms**

Argument	Description
	rtt min/avg/max/mdev = 0.065/0.077/0.091/0.008 ms

Related Commands

The following table summarizes other commands related to the **gigasmart** command:

Task	Command
Displays ARP information for the GigaSMART engine port.	**show gigasmart engine 1/1/e1 arp**
Displays detailed information for the GigaSMART engine port.	**show gigasmart engine 1/1/e1 details**
Displays statistics for the GigaSMART engine port.	**show gigasmart engine 1/1/e1 stats**
Displays ARP information for a specified GigaSMART engine port interface and VLAN ID.	**show gigasmart engine 1/1/e1 interface eth2 vlan 200 arp**
Displays detailed information for a specified GigaSMART engine port interface and VLAN ID.	**show gigasmart engine 1/1/e1 interface eth3 vlan 300 details**
Displays statistics for a specified GigaSMART engine port interface and VLAN ID.	**show gigasmart engine 1/1/e1 interface eth2 vlan 200 stats**
Deletes the default (eth2) interface for Internet connectivity.	**(config) # no gigasmart engine 1/1/e1 interface**
Deletes the specified interface for Internet connectivity.	**(config) # no gigasmart engine 1/1/e1 interface eth3**
Deletes the interface for connectivity with VLAN.	**(config) # no gigasmart engine 1/1/e1 interface eth2 vlan 100**
Deletes all stack port interfaces and the proxy serve profiles that were attached to the GigaSMART engine.	**(config) # no gigasmart engine 1/1/e1 interface all**

gigastream

Required Command-Line Mode = Configure

Use the **gigastream** command to group multiple ports into a logical bundle called a GigaStream.

Starting in software version 4.8, there are two types of GigaStream: regular GigaStream and controlled GigaStream. Controlled GigaStream has GigaStream controlled traffic distribution.

NOTE: Regular GigaStream and controlled GigaStream are not interchangeable. You cannot change the type of GigaStream from regular to controlled on the fly.

All participating ports in any type of GigaStream must be running the same speed and must use the same port type.

Refer to the "*GigaStream*" section in the *GigaVUE Fabric Management Guide* for details on configuring GigaStream.

Weighted GigaStream provides you the ability to distribute traffic to the ports by assigning either an equal weight or a custom weight to the ports. For more information about the Weighted GigaStream, refer to the *"Weighted GigaStream"* section in the *GigaVUE-FM User's Guide*.

The **gigastream** command is also used as part of the configuration of the leaf and spine architecture with multiple paths for achieving high availability in a cluster environment. Refer to the "*Multi-Path Leaf and Spine" section* in the *GigaVUE Fabric Management Guide* for details.

The **gigastream** command has the following syntax:

```
gigastream
 advanced-hash
 alias <alias>
 comment <comment>
  hash-bucket-id <ID/range> <port <port ID/port range>>
  hash-size <1-256>
 port-list <port-list> [params hash advanced]
 hash-weight <weight-list>
 drop-weight <weight>
 rehash
```

The following table describes the arguments for the **gigastream** command:

Argument	Description
advanced-hash	Refer to gigastream advanced-hash.
alias <alias>	Specifies an alias for the GigaStream. The maximum number of characters supported in an alias is 128. Each GigaStream name should be unique across the configured GigaStreams. For example: **(config) # gigastream alias stream1**
comment <comment>	Specifies a unique text string that describes the GigaStream. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # gigastream alias stream1 comment "first GigaStream"**

Argument	Description
hash-bucket-id <ID/range> <port <port ID/port range>>	Specifies the mapping of hash bucket IDs to tool ports. Use this command to add tool ports to the GigaStream and to map the port to a specified hash bucket ID. The number of hash bucket IDs have the same range as the hash size, from 1 to 256. A port ID or range cannot exceed the size specified by the **hash-size** parameter. Ports are specified by their port ID or range of port IDs, so the port ID or range can take a specific hash bucket ID or a range of IDs to be mapped to a port. Examples: **(config) # gigastream alias stream1** **(config gigastream alias stream1) # hash-size 5** **(config gigastream alias stream1) # hash-bucket-id 1..5 port 1/1/x1..x5** or **(config) # gigastream alias stream2** **(config gigastream alias stream1) # hash-size 10** **(config gigastream alias stream2) # hash-bucket-id 1 port 1/2/x1** **(config gigastream alias stream2) # hash-bucket-id 2 port 1/2/x2** **(config gigastream alias stream2) # hash-bucket-id 3 port 1/2/x3** **(config gigastream alias stream2) # hash-bucket-id 4 port 1/2/x4** This parameter applies to controlled GigaStream only. If a hash bucket ID is already configured to a port, for example port x1, if you change it to port x2, then port x2 replaces port x1. At least one hash bucket ID should be mapped to a port before a GigaStream can be attached to a map.
hash-size <1-256>	Specifies the size of the hash bucket for the trunk required. The values are from 1 to 256. This will create a controlled GigaStream of the hash size specified. For example: **(config) # gigastream alias stream1** **(config gigastream alias stream1) # hash-size 5** This parameter applies to controlled GigaStream only. The hash size can be increased or decreased. Refer to the "*Editing Controlled GigaStream"* section in the *GigaVUE Fabric Management Guide*.
port-list <port-list>	Specifies ports. Use one or more of the following separated by commas—no spaces or tabs are allowed: o **port-id**—<bid/sid/pid> o **port-alias**—<port-alias> o **port-list**—<bid/sid/pid_x..pid_y> (range) | <bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list) The **port-list** argument lets you select multiple non-contiguous ports. To enter port IDs in a list, put a comma between each port ID in the list. ForGigaVUE TA Series, GigaVUE-OS on a white box, and GigaVUE-HB1, the slot ID (sid) is always designated as 1 as there is only a single slot. When creating a cross-module GigaStream on a GigaVUE-HC3, use the same syntax, while identifying the correct slot ID (sid) for each port.

Argument	Description
	NOTE: All of the ports in the GigaStream must be running at the same speed—1Gb, 10Gb, 40Gb, or 100Gb and must use the same port types (all g, x, q, or c). All of the ports in the GigaStream must be in the same slot. A stack GigaStream must consist of ports with 10Gb speed or higher. The **<bid/sid/pid_x..pid_y>** format lets you select a series of adjacent ports (for example, **1/5/x4..x6** selects port x4..x6 on slot 5. This parameter does not apply to controlled GigaStream.
params hash advanced	Specifies how traffic should be hashed across member ports in the GigaStream. There is only one option—**advanced**. Use the **gigastream advanced-hash** command to select the criteria for the advanced-hash. Refer to gigastream advanced-hash for details. Refer to the "*Regular GigaStream Failover Protection*" section in the *GigaVUE Fabric Management Guide* for details on how traffic hashing changes depending on the number of ports in the GigaStream.
hash-weight <weight>	Specifies the custom weight for the ports based on which traffic is directed in the GigaStream. You can assign the custom weight in ratio or percentage. Separate each weight in the list with a comma. For example: • Custom weight in ratio: **(config gigastream alias stream1) # port-list 1/1/x1..x4 hash-weight 3,3,2,2** • Custom weight in percentage: **(config gigastream alias stream1) # port-list 1/1/x1..x4 hash-weight 30,30,20,20** To convert the weighted GigaStream to a GigaStream with equal weight, use the following command: **(config) # gigastream alias gs** **(config gigastream alias gs) port-list 1/1/x1..x4** **(config gigastream alias gs) exit**
drop-weight <weight>	Specifies the relative weight for the GigaStream to drop the traffic. For example: **(config gigastream alias stream1) # drop-weight 2**
rehash	Redistributes the hash buckets of the ports in the GigaStream. Use this argument when the weights assigned to the ports do not match the incoming traffic distribution. For example: **(config gigastream alias stream1) # rehash**

gigastream advanced-hash

Use the **gigastream advanced-hash** command to select the criteria for the advanced-hash algorithm. The **advanced-hash** method you specify is used for all types of GigaStream in place on the specified line card or chassis. The advanced-hash configuration affects hashing behavior of the following port types:

- Tool ports
- Hybrid ports
- Circuit ports

The **gigastream advanced-hash** command has the following syntax:

```
gigastream
 advanced-hash
   slot <slot number>
   all
   default
   fields
       ethertype
       gtpteid
      ip6dst
     ip6nextHeader
      ip6src
      ipdst
      ipsrc
      macdst
      macsrc
      mpls
     port6dst
     port6src
     portdst
    portsrc
    protocol
    ingressport
  none
```

The following table describes the arguments for the **gigastream advanced-hash** command:

Argument	Description
slot <slot number>	On the GigaVUE-HC3, and GigaVUE-HC1, GigaStream hashing is per chassis, not per line card. Specify slot cc1 in the CLI command. For the following GigaVUE-HC3 example, the configuration will apply to the chassis: **(config) # gigastream advanced-hash slot 4/cc1 all**
all	Enables all hash criteria fields, including Layer 2, Layer 3, and Layer 4 fields. **Note**: When both Layer 3 (IPv4 or IPv6) and Layer 2 (MAC) fields are enabled for a given GigaStream and there is a mix of Layer 3 and Layer 2 packets, Layer 3 will take precedence. The incremental Layer 3 packets will hash; the incremental Layer 2 packets will not hash.

Argument	Description
default	Sets the advanced-hash algorithm to its default settings. By default, the advanced-hash algorithm includes source/destination IPv4/IPv6 addresses and ports (**ipsrc**, **ipdst**, **ip6src**, **ip6dst**, **protocol**). For example: **(config) # gigastream advanced-hash slot 1/1 default**
fields	Specifies the hash criteria: • **ethertype**—Adds L2 ethertype field. • **gtpteid**—Adds GTP tunnel endpoint identifier. • **ip6dst**—Adds IPv6 destination IP. • **ip6nextHeader**—Adds IPv6 next header field. • **ip6src**—Adds IPv6 source IP. • **ipdst**—Adds IPv4 destination IP. • **ipsrc**—Adds IPv4 source IP. • **macdst**—Adds L2 destination MAC. • **macsrc**—Adds L2 source MAC. • **mpls**—Adds MPLS label (up to three). • **port6dst**—Adds IPv6 destination port. • **port6src**—Adds IPv6 source port. • **portdst**—Adds IPv4 destination port. • **portsrc**—Adds IPv4 source port. • **protocol**—Adds IPv4 protocol. • **ingressport**—Adds ingress port. For example: **(config) # gigastream advanced-hash slot 2/1 fields portsrc portdst** **Note**: Layer 2 hash criteria (**ethertype**, **macdst**, and **macsrc**) are only honored for Layer 2 packets. They are not used to hash TCP/IP packets. Refer to the *"Advanced Hashing"* section in the *GigaVUE Fabric Management Guide* for details. Starting in software version 5.1, **ipsrc**, **ipdst**, **ip6src**, and **ip6dst** fields inside an MPLS tunnel can also be used for hashing across GigaStream ports. Refer to the *"Advanced Hashing with MPLS"* section in the *GigaVUE Fabric Management Guide*. Starting in software version 5.2, GTP TEID can also be used for hashing across GigaStream ports. Refer to the *"Advanced Hashing with GTP TEID"* section in the *GigaVUE Fabric Management Guide*.
none	Clears all fields from the advanced hash. For example: **(config) # gigastream advanced-hash slot 1/1 none**

Related Commands

The following table summarizes other commands related to the **gigastream advanced-hash** command:

Task	Command
Displays regular GigaStream advanced hash fields.	**# show gigastream advanced-hash**
Displays regular GigaStream advanced hash fields for a specified box ID.	**# show gigastream advanced-hash box-id 24**
Displays regular GigaStream advanced hash fields for a specified box ID in table format.	**# show gigastream advanced-hash box-id 24 brief**
Displays regular GigaStream advanced hash fields for a specified slot.	**# show gigastream advanced-hash box-id 24 slot 24/1**
Displays regular GigaStream advanced hash fields for a specified slot in table format.	**# show gigastream advanced-hash box-id 24 slot 24/1 brief**
Displays detailed information on a specified GigaStream.	**# show gigastream alias stream1**
Displays detailed information for all configured GigaStream.	**show gigastream all**
Displays a summary table of all configured GigaStream.	**show gigastream brief**
Deletes a specified GigaStream trunk.	**(config) # no gigastream alias stream1**
Deletes a comment on a specified GigaStream.	**(config) # no gigastream alias stream1 comment**
Deletes the mapping of specified hash bucket IDs to ports for a specified controlled GigaStream. **Note**: If the GigaStream is already attached to a map, the last mapped hash bucket ID cannot be deleted.	**(config) # gigastream alias stream1** **(config gigastream alias stream1) # no hash-bucket-id 4**
Deletes all GigaStream.	**(config) # no gigastream all**
Deletes all GigaStream except stack-link GigaStream.	**(config) # no gigastream all keep-stack**

gigastream variance-threshold

Required Command-Line Mode = Configure

Use the **gigastream variance-threshold** command to configure threshold values based on which traps are triggered in GigaVUE-FM for imbalance traffic distribution with in the ports of GigaStreams. GigaStream imbalance threshold can be configured at the following levels:

- Global level
- Type level
- Per GigaStream Level

Refer to the "GigaStream Imbalance Threshold Notification" section in the .GigaVUE Administration Guide for more details.

The **gigastream variance threshold** command has the following syntax:

gigastream
<global/stack/tool/hybrid/circuit>

alias <alias> variance-threshold <threshold percent value>

The following table describes the arguments for the **gigastream** command:

Argument	Description
alias <alias>	Specifies an alias for the GigaStream. The maximum number of characters supported in an alias is 128. Each GigaStream name should be unique across the configured GigaStreams. For example: **(config) # gigastream alias stream1**
<global/stack/tool/hybrid/circuit>	Type of GigaStream for which variance threshold is to be configured.
variance-threshold <threshold percent value>	variance threshold %

Related Commands

The following table summarizes other commands related to the **gigastream variance-threshold** command:

Task	Command
Displays global level threshold configured	**# showgigastream global-threshold**
Disables the feature at type level.	**# no gigastream <stack/tool/circuit/hybrid> variance-threshold**

gpfcp profile

Required Command-Line Mode = Configure

Use the **gpfcp profile** command to enable the CPN UPN communication on both the CPN and UPN for exchanging the GPFCP packets.

NOTE: The **gpfcp profile** creation must be done on both the CPN and UPN.

The **gpfcp profile** command has the following syntax:

```
gpfcp-profile alias <alias>
profile <add | delete> ip-interface <interface IP> port <port number> type
<control | user> ips <List of IPv4/IPv6 addresses>
```

The following table describes the arguments for the **gpfcp profile** command:

Argument	Description
gpfcp-profile alias <alias>	Specifies the name of the gpfcp profile. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # gpfcp-profile alias gpfcp-profile-1**
profile <add | delete>	Specifies to add a gpfcp profile or delete a gpfcp profile.

add ip-interface <interface IP> port <port number> type <control | user> ips <List of IPv4 or IPv6 addresses>	Specifies to add the attributes of a gpfcp profile. The attributes are as follows:
	▪ **ip-interface <interface IP>** - Specifies the IP address of the interface.
	▪ **port <port number>**- Specifies the port number. The value ranges from 49152 to 65535.
	▪ **type** Specifies any one of the following node type:
	▪ **control** - Specifies the control node type for CPN.
	▪ **user** - Specifies the user node type for UPN.
	NOTE: The control type configured on UPN and the user type configured on CPN.
	▪ **ips <List of IPv4 or IPv6 addresses>** - For 4G or 5G CPN, specify the IP addresses of SGW-U:S1u or PGW-U:s5u or N3 interface.
	▪ **ips <List of IPv4 or IPv6 addresses>** - For 5G UPN, specifies the IP address of the SGW-C:Sxa or PGW-C:Sxb or N4 interface.
	For example:
	`(config) # gpfcp-profile alias gpfcp_upn1_cpn`
	`(config gpfcp-profile alias gpfcp_upn1_cpn) # profile add ip-interface 20.0.0.2 port 50000 type control ips 10.0.0.2,10.0.0.10`
	`(config gpfcp-profile alias gpfcp_upn1_cpn) # exit`

delete ip-interface <interface IP> port <port number>	Specifies to remove the attributes from a gpfcp profile.
	For example:
	`(config) # gpfcp-profile alias gpfcp_upn1_cpn`
	`(config gpfcp-profile alias gpfcp_upn1_cpn) # profile delete ip-interface 20.0.0.2 port 50000`
	`(config gpfcp-profile alias gpfcp_upn1_cpn) # exit`

NOTE: The system supports 256 IPv6 or 256 IPv4 addresses for the gpfcp profile. The IP address in the GPFCP profile can be updated with subnets for GigaVUE-HC3 Gen 2 and Gen3 devices. It is recommended to configure non-overlapping IPs that match the subnet. If overlapping IPs are configured, the system will select the first matching IP for that subnet.

NOTE: Duplicate IP addresses are not allowed in any gpfcp profile's attribute IP address lists. Each IP address within a gpfcp profile must be distinct.

Related Commands

The following table summarizes the commands related to the **gpfcp-profile**command:

Task	Command
Displays a specified gpfcp profile	**show gpfcp-profile alias gpfcp-profile-1**
Displays all the created gpfcp profiles	**show gpfcp-profile all**
GPFCP delay from CPN in msec. Default is 10msec.	**gpfcp-delay interval <number>**

gsgroup

Required Command-Line Mode = Configure

Use GigaSMART groups to manage and budget GigaSMART processing power.

Use the ***gsgroup*** command to create groups of GigaSMART engine ports in a given chassis. In turn, each GigaSMART operation you create must be assigned to a GigaSMART group.

This command does not apply to GigaVUE TA Seriesnodes.

The **gsgroup** command has the following syntax:

```
gsgroup alias <alias>
 port-list <port-list> ingress-rate <50-100>|resource packet-buffer scale <1-2>
 gtp-persistence
        download
        pause
        resume
        upload
reload all|port-list <engine-port-list>
```

GigaSMART engine ports (**e** ports) are numbered with an **e** prefix using **<bid/sid/e1..e2>** nomenclature—**1/1/e1**, for example or **3/2/e1..e2**.

The number of GigaSMART engine ports available in a chassis depends on the number of GigaSMART line cards or modules in the chassis—up to four in the GigaVUE-HC3. The GigaVUE-HC1 has one GigaSMART engine port.

Notes:

- You cannot combine different generations of GigaSMART cards in the same GigaSMART group.
- You should always allow a maximum of 3 minutes time gap when you delete and recreate a gsgroup.

The following table describes the arguments for the **gsgroup** command:

Argument	Description
alias <alias>	Specifies an alias for the GigaSMART group (gsgroup). The maximum number of characters supported in an alias is 128.
port-list <port-list>	Specifies engine ports. Use one or more of the following separated by commas—no spaces or tabs are allowed: • **port-id**—<bid/sid/pid> where pid is e1 or e2 • **port-list**—<bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list) where pid is e1 or e2 The **port-list** argument lets you select multiple non-contiguous ports. To enter port IDs in a list, simply put a comma between each port ID in the list. The **<bid/sid/pid_x..pid_y>** format lets you select a series of adjacent ports (for example, **1/5/x4..x6** selects port x4..x6 on slot 5). For example: **(config) # gsgroup alias gsg1 port-list 1/1/e1**
hash	Allows the GigaSMART group to hash based on the following options: • **advanced**—Hashing based on the fabric advanced-hash fields (refer fabric advanced-hash) • **ipsrc-ipdst**—Hashing based on the two tuple hashing **NOTE:** You can group The front and rear GigaSMART cards in GigaVUE-HC1-Plus by enabling the **ipsrc-ipdst** hash settings.
ingress-rate	Allows to configure the traffic ingress rate limit to GigaSMART group. The range of this ingress rate field is from 50%- 100%. By default the rate would be 100%. **NOTE:** This is applicable only for SMT-HC3-C08, SMT-HC1-S, SMT-HC1A-R GigaSMART cards.
resource packet-buffer scale <1-2>	This command is used to configure packet buffer scale in GigaVUE-HC1 Gen 2 and GigaVUE-HC3 Gen 2 . The default value is 1. In GigaVUE-HC3 Gen 2, you can configure from range 1- 2 scale. If you enter 2 as your scale, ensure to reboot the card after applying the configurations.

Argument	Description
	NOTE: It is recommended to configure your resource packet buffer as 2 scale, only when there is a need to allocate additional ingress packet buffers in theApplication Metadata Intelligence sessions.
gtp-persistence pause <time interval in mins>	This command creates an immediate backup as soon as the pause interval is configured. Immediate backup happens within 1 minute of the pause interval configuration.
gtp-persistence pause <time interval in mins> skip back-up	This command is used to skip immediate backup.
gtp-persistence resume	The resume option is to resume the GTP persistence backup operation.
gsgroup alias <gsgroup name> gtp-persistence upload <URL to remote server> all|slot <slot number> engine <engine number>	The upload option is to upload the persistence backup to a remote server.
gsgroup alias <gsgroup name> gtp-persistence download <url to backup file> all|slot <slot number> engine <engine number>	The download option is to download a backup file from a remote server.
reload all|port-list <engine-port-list>	This command allows to reload either: • **all**-Reload all GigaSMART engines in the gsgroup. • **port-list**- Reload only a subset of GigaSMART engines in the gsgroup. **NOTE:** Supported only on Gen3 GigaSMART cards on GigaVUE HC Series device. **NOTE:** If the group is associated with inline traffic configuration , a warning is displayed indicating that the reload will impact traffic on that group. Proceed only if you have planned for a brief disruption.

upload	Upload persistence data base file to the remote server

Related Commands

The following table summarizes other commands related to the **gsgroup** command:

Task	Command
Displays all GigaSMART groups.	**# show gsgroup**
Displays a specified GigaSMART group.	**# show gsgroup alias gsg1**
Displays all GigaSMART groups.	**# show gsgroup all**
Displays a flow-ops report. Refer to the "*GigaSMART Group Flow Ops Report Statistics Definitions*" section in the *GigaVUE Fabric Management Guide*.	**# show gsgroup flow-ops-report alias gsg1 type <flow-sampling | ssl-decryption | flow-filtering | flow-sip | inline-ssl>**
Displays a particular IMSI associated with the GigaSMART group.	**# show gsgroup flow-whitelist alias gsg1 imsi 318260109318283**
Displays GigaSMART resources for a specified GigaSMART group.	**# show gsgroup gsapp-resource alias gsg1**
Displays GigaSMART resources for all GigaSMART groups. This is applicable only for Gen2 cards. **Note**: Occasionally, the GigaSMART line card or module will need to be reloaded for changes to take effect and to allocate resources accordingly. Reloading also provides applications with contiguous memory. The following message displays at the bottom of the output of the show gsgroup gsapp-resource command when the GigaSMART line card or module needs to be reloaded: ***Resource allocation changes have been made that require GigaSMART card 2/1/1 to be reloaded in order for them to take effect.** When this message is displayed, you cannot change the configuration relating to that application until after the reload. For example, you cannot use the gsop, associated with the gsgroup, in a map. Use the **card slot** command to reload a GigaSMART line card or module.	**# show gsgroup gsapp-resource all**
Displays GTP stateful session recovery information for a specified GigaSMART group.	**# show gsgroup gtp-persistence alias gsg1**
Displays GTP stateful session recovery information for all GigaSMART groups.	**# show gsgroup gtp-persistence all**
Displays statistics for a specified GigaSMART group.	**# show gsgroup stats alias gsg1**
Displays statistics for all GigaSMART groups.	**# show gsgroup stats all**
Deletes a specified GigaSMART group.	**(config) # no gsgroup alias gsg1**
Deletes all GigaSMART groups.	**(config) # no gsgroup all**

gsparams

Required Command-Line Mode = Configure

Use the **gsparams** command to set options for GigaSMART operations on GigaVUE-OS® HC Series nodes.

This command does not apply to GigaVUE TA Seriesnodes.

The **gsparams** command has the following syntax:

```
gsparams gsgroup <GigaSMART group alias>
apptcp-lb <enable | disable>
apptcb-lb <application | control> <broadcast | drop>
cpu utilization type total rising <20-99%>
 dedup-action <count | drop>
   dedup-ip-tclass <ignore | include>
   dedup-ip-tos <ignore | include>
   dedup-tcp-seq <ignore | include>
   dedup-timer <10-500000µs>
   dedup-vlan <ignore | include>
 eflow interval <0- 3600>
 eflow packet-count <integer>
 eflow packet-ratio <0-100>
 eflow logging <enable | disable>
eng-watchdog-timer <<60-600> | disable>
erspan3-timestamp format <gs | none | x12-ts>
   flow-mask <disable | enable <default | offset <0-111> length <1-112>>>
   flow-sampling-device-ip-ranges
       add ip4addr <IP address> <netmask>
     delete <all | <ip-id <1-64>>
   flow-sampling-rate <5-95%>
   flow-sampling-timeout <1-60 min>
   flow-sampling-type <device-ip | device-ip-in-gtp>
 5g-flow timeout <1-6000 in unit of 10 minutes>
    generic-session-timeout <5-600 seconds>
gpfcp-profile add gpfcp_cpn_2upn
gpfcp-delay interval 10
   gtp-control-sample <disable | enable>
  gtp-randomsample <disable | enable>
   gtp-randomsample interval <12-48 hours>
   gtp-flow timeout <1-6000 in the unit of 10 minutes>
   gtp-persistence
     disable
     enable
      file-age-timeout <10-1440>
      interval <10-1440>
      restart-age-time <10-1440>

   gtp-whitelist <add <GTP whitelist file alias> | delete>
   hsm-group
      add <HSM group alias>
      delete
    ip-frag
```

```
      forward <disable | enable>
      frag-timeout <5-180 sec>
      head-session-timeout <15-240 sec>
    lb
      failover <disable | enable>
      failover-thres lt-bw <threshold bandwidth 50-90%> | lt-pkt-rate <packet rate 500-
5000kpps>
      replicate-gtp-c <disable | enable>
      use-link-spd-wt <disable | enable>
    netflow-monitor <add <monitor name> | delete>
  3gpp-node-role [control | user [<1-12000> standalone] | disable][ 5G | LTE ] [<1-10000>]| [<1-
12000> standalone]
  mobility session overload-threshold <<50-90>|disable>
  mobility tunnel overload-threshold <<50-90>|disable>
      resource
      buffer-asf <<2-5> | disable>
      cpu overload-threshold <<50-90> | disable>
    hsm-ssl
      buffer <<1-3> | disable>
      packet-buffer <20-3000>
    packet-buffer overload-threshold <<50-80> | disable>
  inline-ssl
    standalone <enable | disable>
  rtp-port range <1~65535 | x..y>
  sffp-profile <add | delete> <sffp-profile alias>
    sip-portlist <1-65535>
    sip-session timeout <30-300>
    sip-tcp-idle-timeout <20-600>
    sip-whitelist
      add <SIP whitelist file>
      delete
  sip-nat <disable | enable>
    ssl-decrypt
      decrypt-fail-action <drop | pass-tool>
      disable
      enable
      hsm-pkcs11
        dynamic-object <disable | enable>
        load-sharing <disable | enable>
      hsm-timeout <2-5000>
    key-cache-timeout <1-86400>
      key-map
        add service <service alias> key <key alias>
        delete service <<service alias> | all>
      non-ssl-traffic <drop | pass>
      pending-session-timeout <30-120>
      session-timeout <30-3600>
      tcp-syn-timeout <20-600>
      ticket-cache-timeout <1-86400>
        tunnel-health-check
      action <drop | pass>
      disable
```

dstport <destination port for UDP>
enable
interval <5-600>
protocol <icmp | udp>
rcvport <receive port on decapsulation side>
retries <1-5>
roundtriptime <1-4>
srcport <source port for UDP>

NOTE: To enable 3 or 5-tuple hashing, use the command: gsparams gsgroup <GigaSMART group alias> distribution-hash <3tuple | 5tuple>. For Gen3 GigaSMART cards, this command is applicable only for slicing, masking and de-duplication applications.

The following table describes the arguments for the **gsparams** command:

Argument	Description
gsgroup <GigaSMART group alias>	Specifies the alias for this GigaSMART group.
apptcp-lb **<enable | disable>** **<application | control>** **<broadcast | drop >**	Specifies the TCP load balancing options as follows: • **enable**—Enables the application TCP load balancing. • **disable**—Disables the application TCP load balancing. • The default is **disable**. • **Drop**—Packets are sent to the collector if collectore is configured. Drops the packets if the collector is not configured. • **Broadcast**—Broadcast all the packets to all the ports configured as part of all the maps. • **application**—Specifies unknown application message action. • **control**—Specifies the TCP control message action
cpu utilization type total rising <20-99%>	Specifies GigaSMART CPU utilization options as follows: • **rising**—Configures the rising threshold for GigaSMART CPU statistics. The default is 90%. This command sets the rising threshold on the GigaSMART engine port (s), as a percentage from 20 to 99. A CPU utilization alarm can be sent when the rising threshold is exceeded. Alarms are reported to all configured SNMP trap destinations and recorded in the log file. For example: **(config) # gsparams gsgroup gg1 cpu utilization type total rising 95** Refer to the *"GigaSMART CPU Utilization Statistics"* section in the *GigaVUE Fabric Management Guide* for details.
dedup-action <count | drop>	Specifies whether duplicate packets are to be counted or dropped by GigaSMART as follows: • **count**—Counts the duplicate packets, but does not drop them. • **drop**—Drops the duplicate packets.

Argument	Description
	• The default is **drop**. For example: **(config) # gsparams gsgroup gs2port1 dedup-action count** Refer to the *GigaSMART De-duplication* section in the *GigaVUE Fabric Management Guide* for details.
dedup-timer <10-500000µs>	Configures the time interval within which an identical packet will be considered a duplicate. The greater the interval over which traffic can be checked for duplicates, the higher the accuracy of the de-duplication detection and subsequent elimination. The default is 50000µs. For example, if two of the same packets are seen in the specified time interval, the packets will be detected as duplicates. If one packet is seen in the time interval and another packet is seen in a later time interval, the packets will not be detected as duplicates. Retransmissions are not counted as duplicates. For example: **(config) # gsparams gsgroup gs2port1 dedup-timer 55000** Refer to the "*GigaSMART De-duplication*" section in the *GigaVUE Fabric Management Guide*.
dedup-ip-tclass <ignore | include>dedup-ip-tos <ignore | include>dedup-tcp-seq <ignore | include>dedup-vlan <ignore | include>	Fine-tunes how duplicates are detected. You can configure the packet fields that are used to detect duplicates. Different network implementations can change certain packet header fields (for example, the TCP sequence number). If you want to be able to detect duplicates without requiring that these fields match (ToS field, TCP sequence number, VLAN ID), you can disable the corresponding option. The options are as follows: • **dedup-ip-tclass**—Ignores or includes IPv6 traffic class. Use for IPv6. The default is **include**. • **dedup-ip-tos**—Ignores or includes the IP ToS bits when detecting duplicates. Use for IPv4. The default is **include**. • **dedup-tcp-seq**—Ignores or includes the TCP Sequence number when detecting duplicates. The default is **include**. • **dedup-vlan**—Ignores or includes the VLAN ID when detecting duplicates. The default is **ignore**. Include means the field will be included when GigaSMART compares packets. Ignore means the field will be ignored when GigaSMART compares packets. For example: **(config) # gsparams gsgroup gs2port1 dedup-tcp-seq ignore** Refer to the *"GigaSMART De-duplication"* section in the *GigaVUE Fabric Management Guide* for details.
eflow <enable | disable> **eflow interval <0- 3600>** **eflow packet-count**	Specifies the elephant flow options as follows: • **enable**—Enables the detection of elephant flow in the traffic in Application Filtering Intelligence.

Argument	Description
<integer> **eflow packet-ratio <0-100>** **eflow logging <enable | disable>**	• **disable**—Disables the detection of elephant flow in the traffic in Application Filtering Intelligence. The default is **disable**. • **interval**—The interval within which packet-count and packet-ratio for a traffic flow are examined. The interval should be specified in seconds. The range lies between 0 to 3600. Specify the interval as 0 to ignore this parameter. The default value is **2 secs**. **NOTE:** It is recommended to change the interval value as **0** after collecting the required parameters. • **packet-count**— the maximum number of packets to be received by the flow within the given interval to categorize the flow as an elephant flow. The default value is 10,000. • **Packet Ratio**— The percentage concentration of the packets in the flow to the packets seen overall by the gsgroup. Specify 0 to ignore this parameter. The default value is **0**. • **logging**—Prints the parameters of the elephant flow including the 5-tuple information into the GigaSMART logs. The default is **disable**.
eng-watchdog-timer <<60-600> | disable>	Specifies the engine watchdog timer. In rare scenarios, a packet processing core in the CPU of a GigaSMART engine can enter a deadlocked state. The engine watchdog timer detects the issue and reloads the GigaSMART engine after a specified number of seconds. If a core is in a deadlocked state, all packets are dropped. This parameter specifies the engine watchdog timer as follows: • **60-600**—Enables the engine watchdog timer and specifies the number of seconds to wait before restarting the GigaSMART engine. • **disable**—Disables the engine watchdog timer. • The default is **enabled**. The default value for the timer is **60** seconds. For example, to change the engine watchdog timer from the default, specify a value within the range of values: **(config) # gsparams gsgroup gsg1 eng-watchdog-timer 100** For example, to disable the engine watchdog timer: **(config) # gsparams gsgroup gsg1 eng-watchdog-timer disable**
erspan3-timestamp format <gs | none | x12-ts>	Specifies the ERSPAN Type III timestamp trailer format for tunnel decapsulation as follows: • **gs**—Specifies GigaSMART timestamp trailer format. • **none**—Specifies no timestamp trailer. • **x12-ts**—Specifies PRT-H00-X12TS timestamp trailer format. • The default is **none**. For example: **(config) # gsparams gsgroup gsg_erspan erspan3-timestamp format gs** Refer to the *"GigaSMART ERSPAN Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide*.

Argument	Description
flow-mask <disable | enable <default | offset <0-111> length <1-112>>>	Specifies parameters for flow masking to improve GigaSMART packet processing for traffic containing MPLS, L2GRE, or VNTag headers as follows: • **disable**—Disables flow masking. • **enable**—Enables flow masking as follows: • **default**—Specifies a default offset of 14 bytes and a default length of 28 bytes. • **offset**—Specifies the number of bytes from the beginning of the packet to the start of the mask within the packet. The values range from 0 to 111. • **length**—Specifies the number of bytes, following the **offset**, to mask within the packet. The length identifies a traffic flow. The values range from 1 to 112. • The default is **disable**. Masking bytes are limited to 112 bytes from the beginning of the packet. The **offset** plus **length** cannot be greater than 112. Examples: **(config) # gsparams gsgroup gg1 flow-mask enable default** **(config) # gsparams gsgroup gg1 flow-mask enable offset 38 length 8** **(config) # gsparams gsgroup gg1 flow-mask disable** Refer to the *"GigaSMART MPLS Traffic Performance Enhancement" section* in the *GigaVUE Fabric Management Guide* for details.
flow-sampling-device-ip-ranges add ip4addr <IP address> <netmask> | delete <all | <ip-id <1-64>>flow-sampling-rate <5-95%> flow-sampling-timeout <1-60 min> flow-sampling-type <device-ip | device-ip-in-gtp>	Specifies FlowVUE sampling parameters as follows: • **flow-sampling-device-ip-ranges**—Specifies the range of IP addresses that identify a valid device. • **flow-sampling-rate**—Specifies how much GTP traffic from subscribers in the specified IP ranges is sampled. The values range from 5 to 95%. • **flow-sampling-timeout**—Specifies after how much time a flow/device in a sampled IP range is declared idle and is no longer sampled. The values range from 1 to 60 minutes. • **flow-sampling-type**—Specifies whether inner or outer IP addresses are used for FlowVUE sampling as follows: • **device-ip**—Specifies a sample subset of devices based on IP address. • **device-ip-in-gtp**—Specifies a sample subset of devices based on inner IP address in the GTP-u tunnel. For example: **(config) # gsparams gsgroup gsg1 flow-sampling-type device-ip-in-gtp** Use **gsparams** to configure these values and **show gsparams** command to verify these parameters. Refer to the *"GigaSMART FlowVUE"* section in the *GigaVUE Fabric Management Guide* for details and examples on FlowVUE.
5g-flow timeout <1-6000 in	Disconnects a 5G session if it is inactive for the specified timeout value.

Argument	Description
unit of 10 minutes>	The timeout can be configured as an integer from 1 to 6000 , with an incremental value of 10 minutes. The default value is 48 (480 minutes).
generic-session-timeout <5-600 seconds>	Specifies the maximum timeout for a session entry in the session table. This is a global session timeout for the specified GigaSMART group. The values are from **5 to 600 seconds**. The default is **5** seconds. For example: **(config) # gsparams gsgroup gsg1 generic-session-timeout 30** Currently, this timeout only applies to tunnel load balancing for L2GRE tunnel encapsulation. Refer to the "*Load Balancing across Tunnel Endpoints*" section in the *GigaVUE Fabric Management Guide*.
gtp-control-sample <disable | enable>	Enables or disables sampling of GTP control plane (GTP-c) traffic as follows: • **enable**—Specifies that GTP-c packets will be sampled. Only the indicated percentage of the control traffic that matches any of the flow sampling rules will be sent to the tool ports specified in the flow sampling maps. • **disable**—Specifies that GTP-c packets will not be sampled. 100% of the control traffic that matches any of the flow sampling rules will be sent to the tool ports specified in the flow sampling maps. Control traffic for both accepted and rejected sessions will be sent. • The default is **enable**. For example: **(config) # gsparams gsgroup gg1 gtp-control-sample disable** Refer to the "*GTP Flow Sampling*" section in the *GigaVUE Fabric Management Guide*.
gtp-randomsample <disable | enable>	Enables or disables sampling of GTP random sample as follows: • **enable**—Specifies that GTP will be random sampled. • **disable**—Specifies that GTP will not be random sampled. • The default is **disable**.
gtp-randomsample interval <12-48 hours>	Specifies the rotation interval for random sampling. The minimum value is 12 hours and the maximum value of the interval is 48 hours.
gtp-flow timeout <1-6000 in the unit of 10 minutes>	Initiates the disconnection of a GTP session if it is inactive for the specified timeout value. The timeout can be configured as an integer from 1 to 6000, in increments of 10 minutes. The default is 48, which is 480 minutes, which is 8 hours. For example: **(config) # gsparams gsgroup gg1 gtp-flow timeout 60**
gtp-persistence disable enable file-age-timeout <10-1440> interval <10-1440> restart-age-time <10-1440>	Specifies GTP persistence options for recovering sessions from a restart as follows: • **disable**—Disables GTP persistence. • **enable**—Enables GTP persistence. The default is **disable**. • **file-age-timeout**—Specifies the time the backup file is considered to

Argument	Description
	be valid, in minutes. After this timeout expires, the backup file is considered to be stale. The default is 60 minutes. • **interval**—Specifies the time interval between backups, in minutes. The default is 10 minutes. • **restart-age-timeout**—Specifies the time interval following a reboot for aging out sessions, in minutes. This is a shorter interval than that specified using the **gtp-flow timeout**. The **gtp-flow timeout** initiates the disconnection of a GTP session if it is inactive for the specified timeout value, which has a default of 8 hours. The **restart-age-timeout** default is 30 minutes. Examples: **(config) # gsparams gsgroup gsg4 gtp-persistence enable** **(config) # gsparams gsgroup gsg4 gtp-persistence inteval 15**
gtp-whitelist <add <GTP whitelist file alias> | delete>	Specifies the alias of the GTP forward list file to associate with a GigaSMART group (add) or to disassociate from a GigaSMART group (delete). For example: **(config) # gsparams gsgroup gg1 gtpwhitelist add wlf1** **(config) # gsparams gsgroup gg1 gtp-whitelist delete** You can also add multiple alias of the GTP forward list file to associate with a GigaSMART group (add). For example: **(config) # gsparams gsgroup gg1 gtp-whitelist add wlf1** **(config), add w2f2** **(config), add w2f2** **(config)# add w3f3** **(config)**
hsm-group add <HSM group alias> delete	Configures an SSL Hardware Security Module (HSM) group as follows: • **add**—Adds an HSM group to a GigaSMART group. • **delete**—Deletes an HSM group from a GigaSMART group. Only one HSM group can be configured. Examples: **(config) # gsparams gsgroup gg1 hsm-group add hsm-set** **(config) # gsparams gsgroup gg1 hsm-group delete**
ip-frag forward <disable | enable> frag-timeout <5-180 sec> head-session-timeout <15-240 sec>	Specifies IP fragmentation options as follows: • **forward**—Enables or disables IP fragmentation forwarding. • **frag-timeout**—Defines how long non-head fragment packets will stay in the system, from 5 to 180 seconds. • Sometimes non-head fragment packets arrive before their head fragment packet. GigaSMART will keep the packets and wait for their head fragment packet to arrive. If the head fragment packet does not

Argument	Description
	arrive within this timeout value, the fragmented packets will be dropped. • **head-session-timeout**—Defines how long the session entry stays in the system, from 15 to 240 seconds. A session entry is created when a new head fragment packet is received. When subsequent fragment packets arrive, the information in this session will be used to forward the fragmented packets to the same destination as the head fragment packet. For example: **(config) # gsparams gsgroup gsg1 ip-frag frag-timeout 30**
lb failover <disable | enable> failover-thres lt-bw <threshold bandwidth 50-90%> | lt-pkt-rate <packet rate 500-5000kpps> replicate-gtp-c <disable | enable> use-link-spd-wt <disable | enable>	Specifies load balancing options as follows: • **failover**—Enables or disables failover when tool ports are down or thresholds to other tool ports in the load balancing port group are exceeded. The default is disabled. A GigaSMART application failover will occur no more than once in 30 seconds.When the load balance metric is **hashing**, traffic continues to be sent to the hashed tool port until the port goes down. When a tool port goes down, traffic is rehashed to another tool port in the port group. No rehashing is done to the existing session flow when a port comes up, even if it was previously a down port. • **failover-thresh lt-bw**—Specifies failover threshold for Least Bandwidth (lt-bw) and Least Packet Rate (lt-bw-rate) load balancing metrics as follows: • For **lt-bw**, the failover threshold is the percentage of the maximum bandwidth of a tool port. For example, for a 1Gb port, a failover threshold of 90% means that failover to another tool port occurs when the bandwidth reaches 900Mbps. The range is from 50% to 90%. The default is 80%. • For **lt-pkt-rate**, a tool port will failover to another tool port when the packet rate is over the specified threshold, in packets per second. The range is from 500k packets per second (pps) to 5000k (5M). The default is 1M. • **replicate-gtp-c**—Enables or disables replicate GTP control packets (GTP-c). The default is disabled. • **use-link-spd-wt**—Enables or disables weight based on link speed for Weighted Round Robin (wt-round-robin), Weighted Least Bandwidth (wt-lt-bw), Weighted Least Packet Rate (wt-lt-pkt-rate), Weighted Least Connection (wt-lt-conn), and Weighted Least Cumulative Traffic (wt-lt-tt-traffic) load balancing metrics. The default is disabled. When enabled, this parameter ignores the weight configured in the port group.For example, if a port group consists of four tool ports, and one of them is 100Gb and the others are 10Gb, the 100Gb link will be selected about 10 times more than the 10Gb links. For example: **(config) # gsparams gsgroup gsgrp1 lb replicate-gtp-c enable**
netflow-monitor <add	Specifies NetFlow monitor options as follows:

Argument	Description
<monitor name> | delete>	• **add**—Specifies a NetFlow monitor to add by name. • **delete**—Deletes a NetFlow monitor. For example: **(config) # gsparams gsgroup gsg netflow-monitor add mon1** **(config) # gsparams gsgroup gsg netflow-monitor delete**
3gpp-node-role [control | user | disable][5G | LTE] [<1-10000>] | [<1-12000> standalone] **NOTE:** When you change the 3gpp-node-role, make sure to reload the chassis for the changes to take effect.	Specifies the role for both 5G and LTE traffic as follows: • **control** — specifies the control role for both 5g and LTE traffic as follows. The default value is LTE/3G. • **5g**—specifies the control session limit for 5G traffic. The default session limit is 5000, the maximum session limit for GigaVUE-HC3 is 12000, and GigaVUE-OS-HD is 6000. • **lte**—specifies the control role for 3G, LTE traffic. The default session limit is 5000, the maximum session limit for GigaVUE-HC3 is 12000, and GigaVUE-OS-HD is 5000. • **user** — specifies the user role for both 5G and LTE traffic as follows. The default value is LTE/3G. • **standalone**— enables both 4G and 5G capabilities on UPN • **disable**— specifies the pre-CUPS/5G role. For example: **(config) # gsparams gsgroup <alias> cpn 3gpp-node-role control 5G**
mobility session overload-threshold<<50-90>|disable>	Specifies a overload threshold for a mobility session per GigaSMART engine, as follows: • **overload-threshold**—Species an overload threshold from 50 to 90 percent. .The default value is 90. • **disable**—Disables the overload threshold.
mobility tunnel overload-threshold <<50-90>|disable>	Specifies a overload threshold for a mobility tunnel per GigaSMART engine follows: • **overload-threshold**—Species an overload threshold from 50 to 90 percent. . The default value is 90. • **disable**—Disables the overload threshold.
resource buffer-asf <<2-5> | disable>	Allocates application resources for buffering on Application Session Filtering (ASF). This parameter allocates the number of session entries, in millions, as follows: • **2-5**—Allocates from 2 to 5 million session entries for buffer ASF. • **disable**—Removes any configured application resources for buffer ASF. • The default is **disable**. The configured application resources will only be available after the GigaSMART line card or module is rebooted. Refer to the “Displaying

Argument	Description
	GigaSMART Application Resource Usage" section in the *GigaVUE Fabric Management Guide*. The resources for buffer ASF on the GigaVUE-HB1 can only be configured to 2 million sessions. Examples: **(config) # gsparams gsgroup gsgrp1 resource buffer-asf 3** **(config) # gsparams gsgroup gsgrp1 resource buffer-asf disable** Configure the resources for buffer ASF before configuring **apps asf** parameters. Refer to apps asf.
resource cpu overload-threshold <<50-90> | disable>	Specifies an overload threshold for CPU resources for GigaSMART operations as follows: • **overload-threshold**—Species an overload threshold from 50 to 90 percent. • **disable**—Disables the overload threshold. The default is **90**. Examples: **(config) # gsparams gsgroup gsg1 resource cpu overload-threshold 70** **(config) # gsparams gsgroup gsg1 resource cpu overload-threshold disable** **Note**: Utilize the overload threshold for bypassing overload during SSL decryption for inline tools.
resource packet-buffer overload-threshold <<50-80> | disable>	Specifies an overload threshold for packet buffer resources for GigaSMART operations as follows: • **overload-threshold**—Species an overload threshold from 50 to 80 percent. . • **disable**—Disables the overload threshold. • The default is **80**. Examples: **(config) # gsparams gsgroup gsg1 resource packet-buffer overload-threshold 60** **(config) # gsparams gsgroup gsg1 resource packet-buffer overload-threshold disable** **Note**: Utilize the overload threshold for bypassing overload during SSL decryption for inline tools.
inline-ssl **standalone <disable | enable>**	Configures the inline SSL to share resources with other GigaSMART operations as follows: • **disable**—Disables the **standalone** mode, the GigaSMART engine resource allocated for Inline SSL feature is reduced to 50% and the residual GigaSMART engine resource can be configured for other GigaSMART applications. • **enable**—Enables the **standalone** mode, and configures the GigaSMART resources only for the Inline SSL feature. By default the

Argument	Description
	standalone mode is enabled for the Inline SSL feature. Examples: **(config) # gsparams gsgroup gsg1 inline-ssl standalone disable** **(config) # gsparams gsgroup gsg1 inline-ssl standalone enable** The following notification is displayed when the configuration is changed after resource allocation.#Changes take effect after card or system reboot
resource hsm-ssl buffer <<1-3> | disable>	Configures resources for the HSM SSL buffer as follows: • **1-3**—Adds resources for the HSM SSL buffer, from 1 to 3MB, per GigaSMART. • **disable**—Disables the buffer memory resources for the HSM SSL buffer. • The default is **disable**. Examples: **(config) # gsparams gsgroup gsg1 resource hsm-ssl buffer 2** **(config) # gsparams gsgroup gsg1 resource hsm-ssl buffer disable**
resource hsm-ssl packet-buffer <20-3000>	Configures resources for the HSM SSL packet buffer as follows: • **20-3000**—Adds resources for the HSM SSL packet buffer, from 20 to 3000, per connection. • The default is **1000**. Packets are buffered while waiting for the session key. For example: **(config) # gsparams gsgroup gsg1 resource hsm-ssl packet-buffer 600**
rtp-port range <1~65535 | x..y>	Specifies the RTP port or ports for SIP/RTP. You must specify a port or a range of ports, from 1 to 65535. Examples: **(config) # gsparams gsgroup gsg1 rtp-port range 2000** **(config) # gsparams gsgroup gsg1 rtp-port range 20000..40000**
sffp-profile <add | delete> <sffp-profile-alias>	Add or Delete Transport Agent Profile. To configure the sffp profile, refer to sffp profile.
sip-nat <disable | enable>	Configures SIP-NAT feature as follows: • **disable**—Disables the SIP-NAT feature. • **enable**—Enables the SIP-NAT feature.
sip-portlist <1-65535>	Specifies the SIP port list for SIP/RTP. You must specify one or more TCP/UDP ports, from 1 to 65535. Use a comma to separate multiple ports. Examples: **(config) # gsparams gsgroup gsg1 sip-portlist 5060**

Argument	Description
	(config) # gsparams gsgroup gsg1 sip-portlist 5060,5070,5090
sip-session timeout <30-300>	Specifies the SIP session timer for SIP/RTP. This is a SIP session inactivity timer, used to clean up inactive sessions. The range of values is from 30 to 300 seconds. The default is 30 seconds. For example: **(config) # gsparams gsgroup gsg1 sip-session timeout 48**
sip-tcp-idle-timeout <20-600>	Specifies the SIP TCP idle timer for SIP/RTP. The range of values is from 20 to 600 seconds. The default is 20 seconds. For example: **(config) # gsparams gsgroup gsg1 sip-tcp-idle-timeout 30**
sip-whitelist add <SIP whitelist file> delete	Adds or deletes a SIP forward list file for SIP/RTP as follows: • **add**—Adds a SIP forward list. Specify the alias of the SIP forward list file containing IMSIs. • **delete**—Delete the SIP forward list. Examples: **(config) # gsparams gsgroup gsg1 sip-whitelist add whitelist1** **(config) # gsparams gsgroup gsg1 sip-whitelist delete**
ssl-decrypt decrypt-fail-action <drop | pass-tool>	Specifies Passive SSL decryption failover options as follows: • **drop**—Drops all traffic for the session if decryption fails. • **pass-tool**—Passes traffic to a tool port as encrypted packets if decryption fails. • The default is **drop**. An Passive SSL decryption failure occurs when encrypted traffic cannot be decrypted, for example, when an incoming flow exceeds the maximum supported bandwidth. For example: **(config) # gsparams gsgroup grp ssl-decrypt decrypt-fail-action pass-tool**
ssl-decrypt disable enable	Specifies Secure Sockets Layer (SSL) decryption options as follows: • **disable**—Disables Passive SSL decryption on whole GigaSMART group. • **enable**—Enables Passive SSL decryption on whole GigaSMART group. • The default is **disable**. Disable can be used as debugging aid for traffic to bypass the Passive SSL decryption application. For example: **(config) # gsparams gsgroup grp ssl-decrypt enable**
hsm-pkcs11 dynamic-object <disable | enable>	Enables or disables the dynamic object for the HSM PKCS12 file as follows: • **disable**—Disables the HSM PKCS12 dynamic object parameter.

Argument	Description
	• **enable**—Enables the HSM PKCS12 dynamic object parameter. • The default is **enable**. For example: **(config) # gsparams gsgroup grp ssl-decrypt hsm-pkcs11 dynamic-object disable**
hsm-pkcs11 load-sharing <disable | enable>	Enables or disables load sharing for the HSM PKCS12 file as follows: • **disable**—Disables the HSM PKCS12 load sharing parameter. • **enable**—Enables the HSM PKCS12 load sharing parameter. • The default is **enable**. For example: **(config) # gsparams gsgroup grp ssl-decrypt hsm-pkcs11 load-sharing disable**
hsm-timeout <2-5000>	Configures the HSM timeout in milliseconds. The HSM timeout specifies a period of time for the communication between the HSM and GigaSMART. The values are from 2 to 5000ms. The default is 1000ms. For example: **(config) # gsparams gsgroup grp ssl-decrypt hsm-timeout 3600**
ssl-decrypt key-cache-timeout <1-86400> ticket-cache-timeout <1-86400>	Configures the following timeouts used when resuming an Passive SSL decryption session: • **key-cache-timeout**—Configures a timeout for SSL session ID cache, from 1 to 86400 seconds. Applies to SSL 3.0 and TLS 1.x. • **ticket-cache-timeout**—Configures a timeout for TLS ticket cache, from 1 to 86400 seconds. Applies to only TLS 1.x. • The default for each timeout is **10800 seconds**. For example: **(config) # gsparams gsgroup grp ssl-decrypt key-cache-timeout 3600** These timeouts relate to how the SSL server stores the SSL key material and later, how the client resumes a session using the stored key material. The timeouts refer to the two different ways the session can be resumed: using a session key cache or using a TLS ticket cache.
ssl-decrypt key-map add service <service alias> key <key alias> delete service <<service alias> | all>	Specifies Passive SSL decryption and HSM key mappings as follows: • **add**—Adds an SSL decryption or HSM key/service mapping that maps how a key is assigned to a service, which is an IP address of a server. One service can only be mapped to one key on a GigaSMART group. • **delete**—Deletes an SSL decryption or HSM key/service mapping or all key/service mappings. Examples: **(config) # gsparams gsgroup grp ssl-decrypt key-map add service service1 key key1** **(config) # gsparams gsgroup grp ssl-decrypt key-map**

Argument	Description
	delete service service1 The maximum number of key/service mappings is 2000 on GigaVUE-OS HD Series. The maximum number of key/service mappings is 1000 on GigaVUE-HB1. First create an SSL key alias, then a service alias, and then use **key-map** to tie them together. Refer to apps ssl for the commands to create keys, and services, including the default service. A service can be mapped to different keys on different GigaSMART groups.
ssl-decrypt non-ssl-traffic <drop | pass>	Specifies how to handle non-SSL traffic as follows: • **drop**—Drops all non-SSL packets. • **pass**—Passes all non-SSL packets. • The default is **drop**. Use this parameter when Passive SSL decryption sessions have both SSL and non-SSL packets after the SSL 3-way handshake. For sessions that have SSL and non-SSL traffic, for example SMTP with StartTLS, this parameter provides an option to pass the non-SSL traffic in addition to the decrypted traffic. For example: **(config) # gsparams gsgroup grp ssl-decrypt non-ssl-traffic drop**

Argument	Description
ssl-decrypt pending-session-timeout <30-120> session-timeout <30-3600> tcp-syn-timeout <20-600>	Specifies Passive SSL decryption timeout options as follows: • **pending-session-timeout**—Configures a pending session timeout, from 30 to 120 seconds, for when SSL handshake is not completed. The default is 60. • **session-timeout**—Configures a session timeout, from 30 to 3600 seconds, for when the SSL session is established but no packets are received for the session. The default is 300. • **tcp-syn-timeout**—Configures a TCP sync timeout, from 20 to 600 seconds, for when TCP handshake is not completed. The default is 20. For example: **(config) # gsparams gsgroup grp ssl-decrypt session-timeout 90**
tunnel-health-check action <drop | pass> disable dstport <destination port for UDP> enable interval <5-600> protocol <icmp | udp> rcvport <receive port on decapsulation side> retries <1-5> roundtriptime <1-4> srcport <source port for UDP>	Specifies tunnel health check parameters as follows: • **action**—Specifies the tunnel health check action. The values are drop or pass to either drop packets or pass packets if the destination is down. The default is pass. • **disable**—Disables the tunnel health check. The default is disabled. • **dstport**—Specifies the tunnel health check UDP destination port. The range is from 1 to 65535. The default is 54321. The **dstport** and **rcvport** must have the same value. • **enable**—Enables the tunnel health check. The default is disabled. • **interval**—Specifies the tunnel health check interval, which is the frequency of the health check. The range is from 5 to 600 seconds (10 minutes). The default is 600 seconds. • **protocol**—Specifies the tunnel health check protocol. The values are ICMP or UDP. The default is ICMP. The protocols are as follows: • **ICMP**—Health check uses ICMP Echo Request/Reply packets (like ping) • **UDP**—Health check uses UDP packets. • **rcvport**—Specifies the tunnel health check UDP receive port on the decapsulation side. Specify a port that is not in use. The range is from 1 to 65535. The default is 54321. The **rcvport** and **dstport** must have the same value. • **retries**—Specifies the tunnel health check number of retries before declaring the destination down. The range is from 1 to 5. The default is 5. • **roundtriptime**—Specifies the expected maximum round trip time. The range is from 1 to 4 seconds. The default is 1 second. • **srcport**—Specifies the tunnel health check UDP source port. The range is from 1 to 65535. The default is 54321. For example, use the following commands to configure tunnel health check on the encapsulation device: **(config) # gsparams gsgroup grp1 tunnel-health-check enable** **(config) # gsparams gsgroup grp1 tunnel-health-check protocol icmp**

Argument	Description
	(config) # gsparams gsgroup grp1 tunnel-health-check interval 300 **(config) # gsparams gsgroup grp1 tunnel-health-check retries 3** **(config) # gsparams gsgroup grp1 tunnel-health-check action pass** **(config) # gsparams gsgroup grp1 tunnel-health-check srcport 45500** **(config) # gsparams gsgroup grp1 tunnel-health-check dstport 48000** **(config) # gsparams gsgroup grp1 tunnel-health-check roundtriptime 2** For example, when the decapsulation device is a GigaVUE-OS node, use the following commands to configure tunnel health check: **(config) # gsparams gsgroup grp1 tunnel-health-check enable** **(config) # gsparams gsgroup grp1 tunnel-health-check rcvport 48000**

Related Commands

The following table summarizes other commands related to the **gsparams** command:

Task	Command
Displays GigaSMART parameters on all GigaSMART groups.	**show gsparams**
Displays GigaSMART parameters on a specified GigaSMART group.	**show gsparams alias gsg1**
Displays GigaSMART parameters on all GigaSMART groups.	**show gsparams all**

halt

Required Command-Line Mode = Configure

Use the **halt** command to stop all system activities without powering the system down. This is the same as the **reload halt** command. Refer to reload (reboot).

The **halt** command has the following syntax:

halt

gsop

Required Command-Line Mode = Configure

Use the **gsop** command to create GigaSMART operations. GigaSMART operations consist of a name and a supported combination of the available GigaSMART applications you have licensed.

This command does not apply to GigaVUE TA Series nodes.

NOTE: Refer to the "*How to Combine GigaSMART Operations*" section in the *GigaVUE Fabric Management Guide* for details on supported combinations of GigaSMART operations.

NOTE: Refer to the "*Order of GigaSMART Operations*" section in the *GigaVUE Fabric Management Guide* for information on the order in which GigaSMART components are applied in a single operation.

NOTE: Refer to Configure GigaSMART Operations for examples of how to configure the various GigaSMART operations.

NOTE: SNAP/LLC packets are not supported in GigaVUE-HC1-Plus.

The **gsop** command has the following syntax:

```
gsop alias <alias>
 add-header vlan <1-4094>
 apf set
asf enhanced <enhanced asf alias> port-list <gsgroup name>   asf <ASF alias>
  dedup set
 flow-ops <flow-filtering <gtp> | flow-sampling | gtp-flowsample | gtp-whitelist | netflow | sip-flowsample | sip-whitelist | 5g-whitelist | 5g-flowsample>
   inline-ssl <inline SSL profile alias>
lb
 app <asf | gtp | tunnel | 5G> metric <lt-bw | lt-pkt-rate | round-robin | lt-conn | lt-tt-traffic | wt-lt-bw |
 wt-lt-pkt-rate | wt-round-robin | wt-lt-conn | wt-lt-tt-traffic | wt-supi | wt-imsi | hashing <key <imsi | imei | msisdn>>
 app <sip> metric hashing key caller-id
user-name
command-code
session-id
end-to-end id
hop-by-hop-id
app <<5g> metric hashing key <supi | pei | gpsi>>
application-id
        avp-code
             hash <ip-only <inner | outer> | ip-and-port <inner | outer> | 5-tuple <inner | outer> | gtpu-teid>   masking protocol
            enhanced <elb-name>
             none offset <0-9000>
```

```
            ipv4 offset <1-9000>
            ipv6 offset <1-9000>
            udp offset <1-9000>
            tcp offset <1-9000>
            ftp-data offset <1-9000>
            https offset <1-9000>
            ssh offset <1-9000>
           gtp offset <1-9000>
            gtp-ipv4 offset <1-9000>
            gtp-udp offset <1-9000>
            gtp-tcp offset <1-9000>
            <pattern: 1-byte-hex>
            <length: 1-9600>
            sip content-type message/cpim
          port-list <GigaSMART group alias>
          slicing protocol
            none offset <64-9000>
            ipv4 offset <4-9000>
            ipv6 offset <4-9000>
            udp offset <4-9000>
            tcp offset <4-9000>
            ftp-data offset <4-9000>
            https offset <4-9000>
            ssh offset <4-9000>
            gtp offset <4-9000>
            gtp-ipv4 offset <4-9000>
            gtp-udp offset <4-9000>
            gtp-tcp offset <4-9000>
          ssl-decrypt in-port <<ingress port> | any> out-port <<egress port> | auto>
          strip-header
             erspan <0-1023>
            fabric-path <dst-switch-id <0-(2^12-1)>> <src-switch-id <0-(2^12-1)>>
           fm6000-ts <gs | none | x12-ts>
            generic anchor-hdr1 <none | eth | vlan | mpls | ipv4 | ipv6><offset <start | end |
        <integer>>
              <header-count<1-32> [custom-len <1-1500>]<anchor-hdr2 <none | eth | vlan |
        mpls | ipv4 | ipv6 | tcp |         udp | any>>
            gre
            gtp
           isl
           mpls
           mpls+vlan
           vlan <outer | all>
           vntag
           vxlan <0-(2^24-1)>
          trailer
             add crc <enable | disable> <srcid <enable | disable>
            remove
          tunnel-decap type

tunnel-decap type tcp add <listener>
     custom <portsrc <0-65535> portdst <0-65535>>
    erspan flow-id <0-1023>
    gmip portdst <0-65535>
    l2gre key <0~(2^32-1)>
```

vxlan <portsrc <0-65535> portdst <1-65535> vni <0~(2^24-1)>>
tunnel-encap type
gmip <portsrc <0-65535> portdst <0-65535> ipdst <IP address>> [dscp <0-63>] [prec <0-7>] [ttl <1-255>]
l2gre
ip6dst <IPv6 destination address> key <0~(2^32-1)> [dscp <0-63>] [flow-label <0~(2^20-1)>]
[prec <0-7>] [ttl <1-255>]
ipdst <IP address> key <0~(2^32-1)>
pgdst <port group name> key <0~(2^32-1)> session-field <3-tuple-any | 3-tuple-ipv4 | 3-tuple-ipv6 |
5-tuple-any | 5-tuple-ipv4 | 5-tuple-ipv6 | ip-any | ipv4-only | ipv6-only> <inner | outer>
vxlan <portsrc <1-65536> portdst <1-65536> vni <0~(2^24-1) ipdst | ip6dst <ipv4/ipv6 address>>

The following table describes the arguments for the **gsop** command:

Argument	Description
alias <alias>	Specifies the alias for this GigaSMART operation. Use the alias for all management of a GigaSMART operation, including binding it to a map rule.
add-header vlan <1-4094>	Specifies the add-header GigaSMART operation. Packets processed by this GigaSMART operation are tagged with the specified VLAN tag. This feature can be used in conjunction with the **strip-header** operation to differentiate stripped packets using common IP ranges (10.x.x.x or 192.168.x.x) from non-stripped packets in the same IP range. Refer to the "*GigaSMART Header Addition*" section in the *GigaVUE Fabric Management Guide* for details.
apf set	Specifies the APF GigaSMART operation. Packets processed by this operation are evaluated using Adaptive Packet Filtering rules configured with the **map** command's **gsrule** argument. Refer to the "*GigaSMART Adaptive Packet Filtering (APF)*" section in the *GigaVUE Fabric Management Guide*.
asf enhanced <enhanced asf alias> port-list <gsgroup name>	Specifies the enhanced Application Session Filtering (ASF) GigaSMART operation by configuring an alias. Examples: **(config) # gsop alias <gsop alias> apf set asf enhanced <enhanced asf alias> port-list <gsgroup name>** Refer to the *"GigaSMART EnhancedApplication Session Filtering (ASF)* " section in the *GigaVUE Fabric Management Guide* for details.
asf <ASF alias>	Specifies the Application Session Filtering (ASF) GigaSMART

Argument	Description
	operation by configuring an alias. Examples: **(config) # gsop alias gsop2 asf asf2 port-list gsgrp1** **(config) # gsop alias gsop1 apf set asf asf1 port-list gsg1** Refer to the *"GigaSMART Application Session Filtering (ASF) and Buffer ASF"* section in the *GigaVUE Fabric Management Guide* for details.
dedup set	Specifies the de-duplication GigaSMART operation. Packets processed by this operation are analyzed for duplicates. A packet is considered to be a duplicate if its bits are identical to the original packet from Layer 3 (Network layer) onwards, including the payload (differences in Layer 2 are not considered). For example, if two packets are identical except for Time-to-Live (TTL), they will be counted as duplicates. If you use this operation, you can also use **gsparams** to set the following: • the time interval within which an identical packet will be considered a duplicate. • whether duplicates should be counted or dropped. • the packet fields that are used to detect duplicates. For details of the **gsparams** command, refer to gsparams. Refer to the *"GigaSMART De-duplication"* section in the GigaVUE Fabric Management Guide.
flow-ops <flow-filtering <gtp> | flow-sampling | gtp-flowsample | gtp-whitelist | netflow | sip-flowsample | sip-whitelist | | |5g-whitelist |5g-flowsample>	Configures GigaSMART operations as follows: • **flow-filtering <gtp>**—Creates a GigaSMART operation that enables GTP Correlation. Then, create a second level map with a **flowrule** component that specifies which GTP IMSI, IMEI, MSISDN, or version should be filtered from the virtual port to a tool port. For example: **(config) # gsop alias gtpFilter flow-ops flow-filtering gtp port-list gsgrp1** **NOTE:** This command is supported only on C05 cards and not supported on C08 cards. Refer to the *GigaSMART GTP Correlation* section in the *GigaVUE Fabric Management Guide* for details and examples. • **flow-sampling**—Creates a GigaSMART operation that uses FlowVUE to perform subscriber-based IP sampling. Use the **gsparams** command to specify the type of subscribers

Argument	Description
	that are sampled (inner or outer IP addresses), the rate at which they are sampled, the IP ranges themselves, and the timeout values for any idle devices. These settings are unique for each GigaSMART engine group—they cannot be configured on a per-map basis. For example: **(config) # gsop alias gsfvue flow-ops flow-sampling port-list gsgrp2** Refer to the *"GigaSMART FlowVUE"* section in the *GigaVUE Fabric Management Guide* for details and examples. • **gtp-flowsample**—Enables GTP flow sampling. For example: **(config) # gsop alias fs1 flow-ops gtp-flowsample port-list gsgrp3** Refer to the *"GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide* for details and examples. • **gtp-whitelist**—Enables GTP forward listing. For example: **(config) # gsop alias wlf1 flow-ops gtp-whitelist port-list gsgrp4** Refer to the "*GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide* details and examples. • **netflow**—Enables NetFlow generation. For example: **(config) # gsop alias gsop2 flow-ops netflow port-list gsgrp5** Refer to the *"GigaSMART NetFlow Generation"* section in the *GigaVUE Fabric Management Guide* for details and examples. • **sip-flowsample**—Enables SIP flow sampling for SIP/RTP. Examples: **(config) # gsop alias sip-flowsample-no-lb flow-ops sip-flowsample port-list gsg2** **(config) # gsop alias sip-flowsample flow-ops sip-flowsample lb app sip metric hashing key caller-id port-list gsg1** Refer to the *GigaSMART SIP/RTP Correlation* section in the *GigaVUE Fabric Management Guide* for details and examples. • **sip-whitelist**—Enables SIP forward listing for SIP/RTP. Examples: **(config) # gsop alias sip-whitelist-no-lb flow-ops sip-whitelist port-list gsg2**

Argument	Description
	(config) # gsop alias sip-whitelist flow-ops sip-whitelist lb app sip metric hashing key caller-id port-list gsg1
	• **5g-whitelist**—Enables 5g-forward listing. • **5g-flowsampling**—Enables 5g-flowsampling.
inline-ssl <inline SSL profile alias>	Attaches the inline SSL profile to a GigaSMART operation by specifying the alias of the profile. For example: **(config) # gsop alias issl1-gsop inline-ssl sslprofile port-list gsgrp1** Refer to apps inline-ssl for information on profiles for SSL Decryption for inline tools.
lb app <asf | gtp | tunnel> metric <lt-bw | lt-pkt-rate | round-robin | lt-conn | lt-tt-traffic | wt-lt-bw | wt-lt-pkt-rate | wt-round-robin | wt-lt-conn | wt-lt-tt-traffic | | wt-lt-tt-traffic | wt-imsi | wt-supi | hashing [key <imsi | imei | msisdn]> hash <ip-only <inner | outer> | ip-and-port <inner | outer> | 5-tuple <inner | outer> | gtpu-teid> **apps enhanced-lb alias <elb-name> hash-field <add |delete><LIST><inner|outer>exit**	Configures stateful or stateless load balancing. • **app <asf | gtp | tunnel> metric**—Configures the following stateful load balancing metrics for ASF, GTP, or tunnel: ▪ **lt-bw**—least bandwidth. Not supported for tunnel. ▪ **lt-pkt-rate**—least packet rate ▪ **round-robin**—round robin ▪ **lt-conn**—least connection ▪ **lt-tt-traffic**—least cumulative traffic ▪ **wt-lt-bw**—weighted least bandwidth. Not supported for tunnel. ▪ **wt-lt-pkt-rate**—weighted least packet rate ▪ **wt-round-robin**—weighted round robin ▪ **wt-lt-conn**—weighted least connection ▪ **wt-lt-tt-traffic**—weighted least cumulative traffic ▪ **wt-imsi**—weighted IMSI stateful ▪ **wt-supi**—weighted SUPI stateful o **hashing**—hashing (include key). The hashing key only applies to the GTP stateful application. • **hash**—Configures the following stateless load balancing metrics: ▪ **ip-only**—source IP and destination IP addresses ▪ **ip-and-port**—source IP and destination IP addresses, source port and destination port numbers ▪ **5-tuple**—source IP and destination IP addresses, source port and destination port numbers, protocol field in IP header ▪ **gtpu-teid**—GTP-u tunnel ID Also, configures the following field locations for **hash**: · **outer**—first occurrence of header or field · **inner**—second occurrence of header or field

Argument	Description
	NOTE: There is no inner or outer field location for **gtpu-teid**. • **apps enhanced-lb alias <elb-name>**—Configures the following enhanced load balancing hash-field metrics: ▪ **ip** - source IP and destination IP addresses ▪ **l4port** - L4 source port and L4 destination port numbers ▪ **gtpu-teid** - GPRS Tunnel Endpoint Identifier (TEID) Also, configures the following field locations for **hash**: ▪ **outer**—first occurrence of header or field ▪ **inner**—second occurrence of header or field **NOTE:** There is no inner or outer field location for **gtpu-teid**. Refer to the *"GigaSMART Load Balancing"* section in the *GigaVUE Fabric Management Guide* for details.
lbapp <sip> metric hashing key caller-id	Configures stateful load balancing for SIP/RTP. Examples: **(config) # gsop alias sip-fs-lb lb app sip metric hashing key caller-id flow-ops sip-flowsample port-list gsgrp** **(config) # gsop alias sip-wl-lb lb app sip metric hashing key caller-id flow-ops sip-whitelist port-list gsgrp**
app <<5g> metric hashing key <supi | pei | gpsi>>	Configures the stateless load balancing for 5G. You must provide any of the following three keys for multi-hashing: • **SUPI**— Specifies the SUPI. • **pei**—Specifies the pei. • **gpsi**—Specifies the gpsi.
masking protocol none offset <0-9000> ipv4 offset <1-9000> ipv6 offset <1-9000> udp offset <1-9000> tcp offset <1-9000> ftp-data offset <1-9000> https offset <1-9000> ssh offset <1-9000> gtp offset <1-9000> gtp-ipv4 offset <1-9000> gtp-udp offset <1-9000> gtp-tcp offset <1-9000>	Specifies the masking GigaSMART operation. Packets processed by this GigaSMART operation mask the specified field with the supplied pattern. You can specify the field to be masked either in terms of a static, hard-coded offset or by using a relative offset from a specified packet header as follows: • Specify a protocol of **none** with an offset of 0 to indicate the beginning of an Ethernet frame. • Specify a static offset by supplying an offset, length, and pattern. In the following example, masking starts at an offset of 14 bytes by repeating the 0xFF pattern for 88 bytes: **(config) # gsop alias mymask masking protocol none offset 14 pattern ff length 88 port-list GS1**

Argument	Description
<pattern: 1-byte-hex> <length: 1-9600> sip content-type message/cpim	• Use one of the packet header and offset options to specify a relative offset. • **pattern** is the one-byte hexadecimal pattern used for the masking. • **length** specifies how much of the packet from the offset should be masked. Refer to the *"GigaSMART Masking"* section in the *GigaVUE Fabric Management Guide* for details. For SIP/RTP: • **content-type message/cpim** is masking only for UDP. Examples: **(config) # gsop alias sip-content-mask masking protocol sip content-type message/cpim port-list gsgrp** **(config) # gsop alias sip-fs-lb flow-ops sip-flowsample lb app sip metric hashing key caller-id masking protocol sip content-type message/cpim port-list gsgrp** **(config) # gsop alias sip-wl-lb flow-ops sip-whitelist lb app sip metric hashing key caller-id masking protocol sip content-type message/cpim port-list gsgrp**
port-list <GigaSMART group alias>	Specifies the GigaSMART group that will be used to process this GigaSMART operation. Use the ***gsgroup*** command to create groups of GigaSMART engine ports in a given chassis as follows: • Each of the two GigaSMART engine ports in an SMT-HC3-C05 module on GigaVUE-HC3 can process packets at up to 100Gb. • The GigaSMART engine port in the GigaVUE-HC1 node can process packets at up to 20Gb. • The GigaSMART engine port in the GigaVUE-HB1 node can process packets at up to 10Gb. GigaSMART engine ports are numbered with an **e** prefix using **<bid/sid/e1..e2>** nomenclature—**1/1/e1**, for example. **NOTE:** The ports in a GigaSMART group can be on different line cards in the same chassis. However, they must all be on the same chassis. **NOTE:** The slot ID for a GigaVUE-HC1 chassis is fixed at **1**. **NOTE:** . The bay ID for a GigaVUE-HC3, will be 1 to 4, depending on where the module or modules are installed. The number of GigaSMART engine ports available in a chassis will depend on the number of GigaSMART line cards

Argument	Description
	or modules in the chassis—up to four in the GigaVUE-HC3). The GigaVUE-HC1 has one GigaSMART engine port.
slicing protocol none offset <64-9000> ipv4 offset <4-9000> ipv6 offset <4-9000> udp offset <4-9000> tcp offset <4-9000> ftp-data offset <4-9000> https offset <4-9000> ssh offset <4-9000> gtp offset <4-9000> gtp-ipv4 offset <4-9000> gtp-udp offset <4-9000> gtp-tcp offset <4-9000>	Specifies the slicing GigaSMART operation. Packets processed by this GigaSMART operation are sliced after the specified packet header and offset or offset. Refer to the *"GigaSMART Packet Slicing"* section in the *GigaVUE Fabric Management Guide* for details.
ssl-decrypt in-port <<ingress port> | any> out-port <<egress port> | auto>	Specifies the Passive SSL decryption GigaSMART operation as follows: • **in-port**—Specifies the destination port on which to listen. It can be an ingress port number between 1 and 65535 or **any**, which means that traffic will be accepted on any server port from 1-65535. Specifying a port number means that traffic for SSL decryption will only be accepted from that port number. • **out-port**—Specifies the destination port on which to send decrypted traffic. It can be an egress port number between 1 and 65535 or **auto**, which means that the outgoing server port is selected at random or by the following port mapping: Port: in-port—out-port: **IMAP**: 993—143 **POP3**: 995—110 **SMTP**: 465—25 **LDAP**: 636—389 **NNTP**: 563—119 **HTTP**: 443—80 For example: **(config) # gsop alias ssl_dec ssl-decrypt in-port any out-port auto port-list GSGROUP1** Optionally, the de-duplication GigaSMART operation can be applied before SSL decryption. For example: **(config) # gsop alias ssl_dec ssl-decrypt in-port any out-port 333 dedup set port-list gsgrp1** Refer to the *"GigaSMART SSL Decryption for Out-of-Band Tools"* section in the *GigaVUE Fabric Management Guide* for details.
strip-header erspan <0-1023> fabric-path <dst-switch-id <0-(2^12-1)>> <src-switch-id <0-(2^12-1)>> fm6000-ts <gs | none | x12-ts> generic anchor-hdr1 <none | eth | vlan | mpls | ipv4 |	Specifies the strip-header GigaSMART operation to identify and remove the following: • **erspan**—Specifies an ERSPAN flow ID, from 0 to 1023. Use this option to strip an ERSPAN header. Both ERSPAN Type II and Type III headers are supported. A flow ID of zero is a

Argument	Description
ipv6 ><offset <start | end | integer>><header-count<1-32> [custom-len <1-1500>]<anchor-hdr2 <none | eth | vlan | mpls | ipv4 | ipv6 | tcp | udp | any>> **gre gtp isl mpls mpls+vlan vlan <outer | all> vntag vxlan <0-(2^24-1)>**	wildcard value that matches all flow IDs. o **fabric-path**—Specifies packets matching a destination switch ID and source switch ID, for Cisco FabricPath headers. The **dst-switch-id** and **src-switch-id** attributes are mandatory. Enter a value from 0 to 4095 (<0-(2^12-1)>) for a 12-bit switch ID. Enter 0 to strip all switch IDs. • **fm6000-ts**—Specifies how to handle the FM6000 timestamp, as follows: o **gs**—Specifies to strip the FM6000 timestamp, convert to UTC, and add the UTC timestamp to the GigaSMART trailer. o **none**—Specifies to strip the FM6000 timestamp. o **x12-ts**—Specifies to strip the FM6000 timestamp, convert to UTC, and add the UTC timestamp to the PRT-H00-X12TS trailer. • **generic**—Specifies to strip any arbitrary header from the packet, by using the offset and the length of the header. o **anchor-hdr1**—Specifies the protocol from where GigaSMART should start stripping the header. The protocols supported are as follows: ▪ **none** —starts stripping the header from the start of the packet ▪ **eth**—starts stripping the packet from Ethernet header ▪ **vlan**—starts stripping the packet from VLAN header ▪ **mpls**—starts stripping the packet from MPLS header ▪ **ipv4**— starts stripping the packet from IPv4 header ▪ **ipv6**— starts stripping the packet from IPv6 header o **offset**—Specifies exactly from which end of the first anchor header the stripping operation should start. The following offset can be specified: ▪ **start**—starts stripping the packet from the left end of the first anchor header. ▪ **end**—starts stripping the packet from the right end of the first anchor header. ▪ **<integer>**—starts stripping the packet from the specified integer offset of the first anchor header. The integer value depends on the anchor-hdr1. o **header-count—**Specifies how many headers from the offset GigaSMART should remove. The header count value can be 1 to 32. o **anchor-hdr2—**Specifies the protocol that should become the next header after the stripping operation is complete. The protocols supported are as follows: ▪ **none** —specifies that the next possible header is

Argument	Description
	none ■ **eth**—specifies that the next possible header is Ethernet ■ **vlan**—specifies that the next possible header is VLAN ■ **mpls**—specifies that the next possible header is MPLS ■ **ipv4**— specifies that the next possible header is IPv4 ■ **ipv6**— specifies that the next possible header is IPv6 ■ **tcp**—specifies that the next possible header is TCP ■ **udp**—specifies that the next possible header is UDP ■ **any**—specifies that the next possible header can be any of the above headers in the packet.
strip-header **NOTE:** In multiple GRE header encapsulation, only the first GRE header is stripped.	• **gre**—Specifies outer IPv4/GRE headers. • **gtp, isl**—Specifies header and trailer of ISL or GTP-encapsulated packets (tunneled packets). • **mpls, mpls+vlan, vlan, vntag, vxlan**—Specifies MPLS headers, VLAN headers, MPLS and VLAN headers, VN-Tag, or VXLAN headers. For VXLAN headers, you can either strip all VXLAN packets with a matching header value or, alternatively, enter a value of 0 for the VXLAN ID to strip the headers from all VXLAN packets. The syntax is as follows: **(config) # gsop alias <alias> strip-header vxlan <0-(2^24-1)>** Refer to the "*GigaSMART Header Stripping*" section in the *GigaVUE Fabric Management Guide* for details. **NOTE:** GTP tunnel stripping does not support stripping packets with an outer IPv6 header. However, using Generic Header Stripping you can strip GTP when an outer IPv6 header is present.
trailer add crc <enable | disable> <srcid <enable | disable> **trailer remove**	Specifies the trailer GigaSMART operation and whether to include or remove the GigaSMART trailer with this operation. This command is specific to Gen 2 GigaSMART module and Gen 3 GigaSMART module. The arguments are as follows: • **crc**—Specifies whether to include the original packet's CRC as a field in the trailer. **NOTE:** The modified packet's actual CRC is always recalculated to reflect its new length. This argument only specifies whether to include the original packet's CRC as a field in the trailer. • **srcid**—Specifies whether to include the Source ID field as a field in the trailer. The Source ID field indicates the port where a packet entered the Gigamon Deep Observability

Argument	Description
	Pipeline. • **remove**—Specifies the trailer to remove. It is useful in situations where you have **cascade connections**—a tool port receiving packets with a GigaSMART trailer is physically cabled to a GigaVUE-OS network port, sending the packets received on the tool port back into aGigaVUE HC Series node. In cases like these, you may want to remove the GigaSMART trailer before the packets are forwarded to other tools. Refer to the *"Using GigaSMART Trailers"* section in the *GigaVUE Fabric Management Guide* for details.
tunnel-decap type tls-pcapng decap-key <key alias> add <listener alias>	Specifies the tunnel-decap GigaSMART operation type tls-pcapng to use in conjunction with a tunneled network port to configure the receiving end of a tunnel. You can use a tunnel-decap port on a GigaVUE HC Series node to receive and decapsulate tunneled traffic. Specify the type of tunnel in the command as follows: • **decap-key** : The RSA key to be validated in the decapsulation. Only packets matching the key values will be decapsulated. • **listener alias** : Add the Listener profile.
tunnel-decap type custom <portsrc <0-65535> portdst <0-65535>> erspan flow-id <0-1023> gmip portdst <0-65535> l2gre key <0~(2^32-1)> vxlan <portsrc <0-65535> portdst <1-65535> vni <0~(2^24-1)>>	Specifies the tunnel-decap GigaSMART operation to use in conjunction with a tunneled network port to configure the receiving end of a tunnel. You can use a tunnel-decap port on aGigaVUE HC Series node to receive and decapsulate tunneled traffic. Specify the type of tunnel in the command as follows: • **custom**—Specifies custom tunnel termination at GigaSMART, with a source and destination port in the range from 0 to 65535. When the source or destination port is 0, the packet will not check for the presence of a Layer 4 (L4) header or will not be validated against the L4 port if present in packet. Refer to the *"GigaSMART Custom Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide*. • **erspan flow-id**—Specifies an ERSPAN flow ID, from 0 to 1023. Use this option when decapsulating traffic received over a Cisco-standard ERSPAN tunnel. Both ERSPAN Type II and Type III headers are supported. A flow ID of 0 decapsulates all ERSPAN tunnel traffic regardless of flow ID. Refer to the "*GigaSMART ERSPAN Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide* for details. • **gmip portdst**—Specifies the UDP port, from 0 to 65535, on which the tunnel network port on the receiving GigaVUE HC Series is listening. Use this option when decapsulating traffic from a GigaSMART-enabled node. The setting must

Argument	Description
	match the configuration of the **portdst** configured on the sending end of the tunnel. Refer to the "*GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel)*" section in the *GigaVUE Fabric Management Guide* for details. • **l2gre key**—Specifies the type of tunnel to decapsulate the packet and the GRE key to be validated in the GRE decapsulation. Only packets matching the key values will be decapsulated. Other packets will be dropped.The key is a 32-bit value. The range is from zero (0) to 2^{32} - 1. If you configure the key as 0, the key field is not validated, that is, the key field in the packet can have any value or no value. Refer to the "*GigaSMART Layer 2 GRE Tunnel Encapsulation/Decapsulation"* section in the *GigaVUE Fabric Management Guide* for details. • **vxlan**—Specifies VXLAN tunnel termination at GigaSMART, with a source UDP port in the range from 0 to 65535, a destination port in the range from 1 to 65535, and a VXLAN Network Identifier (VNI) in the range from 0 to 2^{24} - 1. When the source port is 0, the packet will not be validated against the L4 source port. When the VNI is 0, all VXLAN identifiers will be stripped. Refer to the *"GigaSMART VXLAN Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide* for details. Examples: **(config) # gsop alias tun_decap tunnel-decap type l2gre key 12314 port-list gsport1** **(config) # gsop alias de_tunnel1 tunnel-decap type custom portsrc 100 portdst 4789 port-list gsgroup1** (config) # gsop alias de_tunnel2 tunnel-decap type vxlan portsrc 100 portdst 100 vni 10 port-list gsgroup1
tunnel-encap type gmip <portsrc <0-65535> portdst <0-65535> ipdst <IP address>> [dscp <0-63>] [prec <0-7>] [ttl <1-255>]	Specifies the tunnel-encap type, GMIP, to use in conjunction with a network port that is associated with an IP interface to configure the sending end of a tunnel. GigaSMART tunnels send traffic arriving from a GigaSMART-enabled node over the Internet to a second GigaSMART-enabled node where the traffic is decapsulated and made available to local packet distribution. The arguments are as follows: • **portsrc**—Specifies the UDP port, from 0 to 65535, used in the headers of tunneled packets sent to the destination. • **portdst**—Specifies the UDP port, from 0 to 65535, on which the network port that is associated with an IP interface and residing on the destination node is listening. The **portdst** must match the configuration of the corresponding tunnel-decap operation's **portdst**.

Argument	Description
	• **ipdst**—Specifies the IP address of the port, that is associated with the IP interface alias, on the destination GigaSMART-enabled node. By default, the tunnel-encapsulation application copies the ToS byte from the inner packet to the header of the tunnel packet, ensuring the same values. Similarly, TTL is automatically set to ensure delivery of packets from the sending node to the receiving node. However, you can use the following options to change the QoS assigned to tunneled packets: • **dscp**—Specifies a decimal DSCP value from 0 to 63 to be used in the ToS byte of the outer headers of tunneled packets. The default is 0. • **prec**—Specifies a decimal precedence value from 0 to 7 to be used in the ToS byte of the outer headers of tunneled packets. The default is 0. • **ttl**—Specifies the TTL value. If you find that tunneled packets are expiring in transit from source to destination, you can increase the TTL value used in the outer headers of tunneled packets with this option. Increasing the TTL allows tunneled packets to transit more hops before expiring (each hop decrements a packet's TTL by one). The values are from 1 to 255. The default is 255. Refer to the *GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel)* section in the *GigaVUE Fabric Management Guide* for details.
tunnel-encap type l2gre ip6dst <IPv6 destination address> key <0~(2^32-1)> [dscp <0-63>] [flow-label <0~(2^20-1)>] [prec <0-7>] [ttl <1-255>]	Specifies the type of tunnel, Layer 2 GRE, to encapsulate the packet. The arguments are as follows: • **ip6dst**—Specifies the IPv6 destination address to be used in the encapsulation. • **key**—Specifies the key to be added in the GRE encapsulation. The key is a 32-bit value. The range is from zero (0) to 2^{32} - 1. If you configure the key as 0, the GRE header will not carry the key field and the key bit will be set to 0. Use the same GRE key at tunnel-encap and tunnel-decap ends for successful tunneling. **NOTE:** Use the same GRE key at tunnel-encap and tunnel-decap ends for successful tunneling. • **dscp**—Specifies a decimal DSCP value from 0 to 63 to be used in the ToS byte of the outer headers of tunneled packets. The default is 0. • **flow-label**—Specifies a label to identify a particular flow. The flow label is a 20-bit value. The range is from zero (0) to 2^{20} - 1. • **prec**—Specifies a decimal precedence value from 0 to 7 to be used in the ToS byte of the outer headers of tunneled

Argument	Description
	packets. The default is 0. • **ttl**—Specifies the TTL value. If you find that tunneled packets are expiring in transit from source to destination, you can increase the TTL value used in the outer headers of tunneled packets with this option. Increasing the TTL allows tunneled packets to transit more hops before expiring (each hop decrements a packet's TTL by one). The values are from 1 to 255. The default is 255. For example: **(config) # gsop alias gs_tunnel tunnel-encap type l2gre ip6dst 2001::3 key 5 flow-label 2452 ttl 25 dscp 62 prec 3 port-list gsop1**
tunnel-encap type l2gre ipdst <IP address> key <0~(2^32-1)> pgdst <port group name> key <0~(2^32-1)> session-field <3-tuple-any | 3-tuple-ipv4 | 3-tuple-ipv6 | 5-tuple-any | 5-tuple-ipv4 | 5-tuple-ipv6 | ip-any | ipv4-only | ipv6-only> <inner | outer>	Specifies the tunnel-encap type, Layer 2 GRE, to use in conjunction with a tunneled network port to configure the sending end of a tunnel. The arguments are as follows: • **ipdst**—Specifies the IP address (IPv4) of the port, that is associated with the IP interface alias, on the destination GigaSMART-enabled node. • **pgdst**—Specifies the port group destination. • **key**—Specifies the GRE key that identifies the source of the tunnel. The key is a 32-bit value. The range is from zero (0) to 2^{32} - 1. If you configure the **key** as 0, the GRE header will not carry the key field and the key bit will be set to 0. Use the same GRE key at tunnel-encap and tunnel-decap ends for successful tunneling. • **session-field**—Specifies the attributes of a session field for stateful load balancing as follows: ▪ **3-tuple-any**—Specifies any IPv4/IPv6 3-tuple-based session. ▪ **3-tuple-ipv4**—Specifies an IPv4 3-tuple-based session. The hash value is extracted from the combination of ipv4-src, ipv4-dst, ipv4-protocol. ▪ **3-tuple-ipv6**—Specifies an IPv6 3-tuple-based session. The hash value is extracted from the combination of ipv6-src, ipv6-dst, ipv6-protocol. ▪ **5-tuple-any**—Specifies any IPv4/IPv6 5-tuple-based session. ▪ **5-tuple-ipv4**—Specifies an IPv4 5-tuple-based session. The hash value is extracted from the combination of ipv4-src, ipv4-dst, l4port-src, l4port-dst, ipv4-protocol. ▪ **5-tuple-ipv6**—Specifies an IPv6 5-tuple-based session. The hash value is extracted from the combination of ipv6-src, ipv6-dst, l4port-src, l4port-dst, ipv6-protocol. ▪ **ip-any**—Specifies any IPv4/IPv6-based session.

Argument	Description
	▪ **ipv4-only**—Specifies an IPv4-only-based session. The hash value is extracted from the combination of ipv4-src, ipv4-dst ▪ **ipv6-only**—Specifies an IPv6-only-based session. The hash value is extracted from the combination of ipv6-src, ipv6-dst In addition, for all session fields, specify the following: · **outer**—the first occurrence of the header in the packet · **inner**—the second occurrence of the header in the packet Examples: **(config) # gsop alias tun_encap tunnel-encap type l2gre ipdst 1.1.1.1 key 123214 port-list gsport1** **(config) # gsop alias gsop1 tunnel-encap type l2gre pgdst pg1 key 10 session-field 5-tuple-ipv4 outer lb app tunnel metric round-robin port-list gsport1** **(config) # gsop alias gsop2 tunnel-encap type l2gre pgdst pg1 key 123 lb hash 5-tuple outer port-list gsport1** Refer to the *GigaSMART Layer 2 GRE Tunnel Encapsulation/Decapsulation* section in the *GigaVUE Fabric Management Guide* for details.
tunnel-encap type vxlan <portsrc <1-65536> portdst <1-65536> vni <0~(2^24-1) ipdst | ip6dst <ipv4/ipv6 address>	Specifies the type of tunnel, VxLAN, to encapsulate the packet. The arguments are as follows: • **portsrc**— Specifies the UDP port, from 1 to 65536, used in the headers of tunneled packets sent to the destination. • **portdst**— Specifies the UDP port, from 1 to 65536, on which the network port that is associated with an IP interface and residing on the destination node is listening. The portdst must match the configuration of the corresponding tunnel-decap operation's portdst.. • **vni**—Specifies the VXLAN Network Identifier (VNI) in the range from 0 to 2^{24} - 1 • **ipdst**— Specifies the IP address or IPv6 address of the port, that is associated with the IP interface alias, on the destination GigaSMART-enabled node. For example: **(config) # gsop alias en-tunnel tunnel-encap type vxlan portsrc 100 portdst 100 vni 10 ip6dst 2001:0db8:85a3:0000:0000:8a2e:0370:7334**
tunnel-encap type tls-pcapng **exporter <exporter alias>|exporter-group <exporter group alias>** **port-list <gsgroup-alias>**	Specifies the type of tunnel, tls-pcapng, to encapsulate the packet. The arguments are as follows: • **exporter alias**—Specify the Exporter alias. • **exporter group alias**— Specify the Exporter Group.

Argument	Description
	• **gsgroup-alias** — Specifies the alias for this GigaSMART group.

Related Commands

The following table summarizes other commands related to the **gsop** command:

Task	Command
Displays all GigaSMART operations.	**# show gsop**
Displays a specified GigaSMART operation.	**# show gsop alias gsop1**
Displays all GigaSMART operations.	**# show gsop all**
Displays GigaSMART operations by application.	**<add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt>**
Displays statistics for all GigaSMART operations.	**# show gsop stats**
Displays statistics for a specified GigaSMART operation.	**# show gsop stats alias gsop1**
Displays IP fragmentation statistics for a specified GigaSMART operation.	**# show gsop stats alias gsop1 ip-frag**
Displays statistics for all GigaSMART operations.	**# show gsop stats all**
Displays detailed statistics for all GigaSMART operations.	**# show gsop stats all detail**
Displays statistics of all GigaSMART operations using a particular GigaSMART application.	**by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt>**
Displays statistics in a particular GigaSMART group.	**by-gsgroup gsg1**
Deletes a specified GigaSMART operation.	**(config) # no gsop alias gsop1**
Deletes all GigaSMART operations.	**(config) # no gsop all**

hb-profile

Required Command-Line Mode = Admin

Use the **hb-profile** command to configure a heartbeat profile, which is a group of attributes that you can apply to an inline tool to configure the heartbeat operation of the inline tool.

For a negative heartbeat profile, refer tonhb-profile.

Also refer toinline-tool for information on enabling heartbeat and associating a heartbeat profile with an inline tool.

This command is only applied to GigaVUE HC Seriesnodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader.

If the inline tool through which the heartbeat packets are passed is expecting IPv6 traffic exclusively, you must select a custom heartbeat packet.

The maximum number of heartbeat profiles supported is equal to the maximum number of inline tools, which is 48 on the GigaVUE-HC3, and 8 on the GigaVUE-HC1.

This command is used in the inline bypass solutions described in Configure Inline Bypass Solutions and in the flexible inline arrangements described inConfigure Flexible Inline Arrangements.

The **hb-profile** command has the following syntax:

```
hb-profile <alias <alias> | default>
  custom-packet <URL of PCAP file | none>
  direction <a-to-b | b-to-a | bi-directional>
  packet-format <arp | custom>
  period <period>
  recovery-time <recovery time>
  retry-count <retry count>
  timeout <timeout>
```

The following table describes the arguments for the **hb-profile** command.

Argument	Description
<alias <alias> | default>	Specifies the name of the heartbeat profile. Use the alias to configure a heartbeat profile to associate with an inline tool. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. The default alias is default. You can configure a default heartbeat profile, using the keyword **default**, for example: **(config) # hb-profile alias default** **(config hb-profile alias default) #** All the parameters for **hb-profile** have default values, so you can also configure a default heartbeat profile by providing an alias for it. For example: **(config) # hb-profile alias hb_5** **(config hb-profile alias hb_5) #**
custom-packet <URL of PCAP file | none>	Specifies the URL of the custom heartbeat packet, downloaded from a PCAP file, or **none**. The default is **none**, which means a standard ICMP ARP packet will be used as a heartbeat packet. Custom heartbeat packets are needed in situations in which inline tools do not reliably pass standard ARP packets. For example, if an inline tool is configured to pass only IPv6 traffic, an ICMPv6 ARP packet might be appropriate. The size of a custom heartbeat packet must be less than 128 bytes. If the PCAP file contains several packets, the first packet present in the file is taken as the heartbeat packet. For example:

Argument	Description
	(config hb-profile alias hb_5) # custom-packet http://1.1.1.1/tftp/temp/ARPPackets.pcap The supported formats for download are HTTP, HTTPS, FTP, TFTP, SCP, and SFTP. Use the **show hb-profile** command to display the name of the PCAP file from which the custom heartbeat packet was imported. The PCAP file must be valid before the heartbeat profile can be associated with an inline tool. If you are specifying a custom heartbeat packet as well as a negative heartbeat packet, do not use the same PCAP file for both. Refer to the **hb-ip-addr-a** and **hb-ip-addr-b** parameters under inline-tool.
direction <a-to-b | b-to-a | bi-directional>	Specifies the direction of the heartbeat packet as follows: • **a-to-b**—Specifies from side A to side B of the inline tool. • **b-to-a**—Specifies from side B to side A of the inline tool. • **bi-directional**—Specifies both directions. The default is **bi-directional**. For example: **(config hb-profile alias hb_5) # direction a-to-b**
packet-format <arp | custom>	Specifies the format of the heartbeat packet as follows: • **arp**—Specifies that a standard ICMP ARP packet is used as the heartbeat packet. • **custom**—Specifies that a custom packet is used as the heartbeat packet. For a custom packet, you must also provide the URL of the custom heartbeat packet. The default is **arp**. For example: **(config hb-profile alias hb_5) # packet-format custom**
period <period>	Specifies the period of the heartbeat packet. This is the number of milliseconds between sending subsequent heartbeat packets. The range is from 30 to 5000 milliseconds. The default is 1000 milliseconds. For example: **(config hb-profile alias hb_5) # period 500**

Argument	Description
recovery-time <recovery time>	Specifies the recovery time of the heartbeat packet. This is the minimum number of seconds of successfully received packets to declare that the inline tool is up. The range is from 5 to 60 seconds. The default is 30 seconds. For example: **(config hb-profile alias hb_5) # recovery-time 50**
retry-count <retry count>	Specifies the retry count of the heartbeat packet. This is the number of consecutive timed-out heartbeat packets at which the system will trigger a failover condition. The range is from 0 to 5. The default is 3. For example: **(config hb-profile alias hb_5) # retry-count 2**
timeout <timeout>	Specifies the timeout of sending the heartbeat packet. This is the number of milliseconds allowed for a heartbeat packet between sending and receiving. The range is from 20 to 1000 milliseconds. The default is 500 milliseconds. For example: **(config hb-profile alias hb_5) # timeout 1000**

Related Commands

The following table summarizes other commands related to the **hb-profile** command:

Task	Command
Displays all heartbeat profiles.	**# show hb-profile**
Displays a specified heartbeat profile.	**# show hb-profile alias hb_5**
Displays the default heartbeat profile.	**# show hb-profile alias default**
Displays all heartbeat profiles.	**# show hb-profile all**
Deletes a specified heartbeat profile.	**(config) # no hb-profile alias hb_5**
Deletes a custom packet associated with a specified heartbeat profile.	**(config) # no hb-profile alias hb_5 custom-packet**
Deletes the default heartbeat profile.	**(config) # no hb-profile alias default**
Deletes a custom packet associated with the default heartbeat profile.	**(config) # no hb-profile alias default custom-packet**
Deletes all heartbeat profiles.	**(config) # no hb-profile all**

header-strip

Required Command-Line Mode = Configure

Use the **header-strip** command to configure the Header Stripping protocol for a chassis. For more information about the Header Stripping protocol, refer to the *"About VXLAN Header Stripping"* and *"About MPLS Header Stripping"* sections in the *GigaVUE-FM User's Guide*.

The **header-strip** command has the following syntax:

header-strip box id <box id|all> vxlan aging-interval <300 to 1000000 seconds and 0 to disable>

mpls add <mpls labels>

mpls delete <mpls labels | all>

The following table describes the arguments for the **header-strip** command.

Argument	Description
box id | all	Specifies the boxes on which to configure the settings.
vxlan aging-interval <300 to 1000000 seconds and 0 to disable>	Configures the VXLAN aging interval in seconds to refresh the traffic intelligence capability for the chassis. The valid range is 300 to 1000000 seconds. The default is 0, which disables the VXLAN ageing interval. For example: **(config) # header-strip box id 1 vxlan aging-interval 300**
mpls add <mpls labels>	Configures the MPLS labels for the specified box ID. The valid range is 2 – 1048575. For example: **(config) # header-strip box id 1 mpls add 1200..1300,4..765,3000**
mpls delete <mpls labels | all>	Deletes the specified MPLS labels for the box ID. For example: **(config) # header-strip box id 1 mpls delete 1250..1290** Deletes all the MPLS labels configured for the box ID. For example: **(config) # header-strip box id 1 mpls delete all**

Related Commands

The following table summarizes other commands related to the **header-strip** command:

Task	Command
Displays the VXLAN aging interval configured for the specified box ID.	**# show header-strip box-id 1 aging-interval**
Displays the statistical data such as the list of VXLAN IDs, number of incoming packets that match the VXLAN IDs, and the incoming bytes.	**# show header-strip box-id 1 stats vxlan-id**

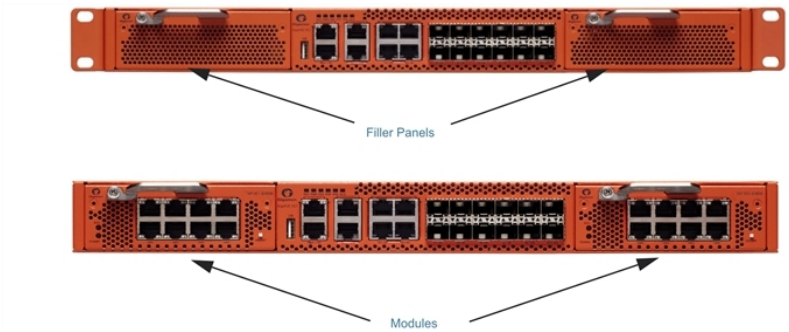
Task	Command
Displays the statistical data such as the list of destination IP addresses, number of incoming packets that match the destination IP addresses, and the incoming bytes.	**# show header-strip box-id 1 stats dest-ip**
Displays the list of MPLS labels added for all the box IDs.	**# show header-strip box-id all** -or- **# show header-strip box-id all mpls**

help

Required Command-Line Mode = Standard or Higher

Use the **help** command to see a quick summary of how to work with online help in the GigaVUE-OS, including description of common conventions.

For example:



hrot

Required Command-Line Mode = Configure

Use the **hrot** command to seek the root of trust in the system with the help of secure micro-controller, highly resistant to attacks and provide most robust form of IP security with multiple level of key verification during power on. Issue the **show version** command to display the HRoT related configuration.

The **hrot** command for GigaVUE-HC1-Plus, and GigaVUE-HCT has the following syntax:

hrot upgrade

```
primary-firmware
key-hash
bios-hash
```

hrot logging

```
err
warning
notice
info
```

The following table describes the arguments for the **hrot** command :

Argument	Description
hrot upgrade primary-firmware| key-hash| bios-hash	Configures HRoT in the firmware. **primary-firmware**-Upgrade HRoT primary-firmware. **key-hash**-Upgrade HRoT primary firmware's key-hash. **bios-hash**-Upgrade HRoT primary firmware's bios-hash.
hrot logging err|warning|notice| info	This would display the HRoT logging states of the firmware such as: **err**-It means device will send HRoT related error messages while boot-up, if occurs. **warning**-It means device will send HRoT related warning messages while boot-up, if occurs. **notice**- It means the device is under normal condition and will send HRoT related significant condition messages while boot-up. **info** -It means that the device will display detailed HRoT related informational messages while boot-up. It will also display error, warning and notice related logs as well. • By default the device will send **notice**logs. • If you modify the command value the device should be reloaded for the command to be in effect • If you use the command **warning** the device will send both warning and error logs.

hostname

Required Command-Line Mode = Configure

Use the **hostname** command to specify the GigaVUE HC Series node's hostname. The hostname will appear in the system prompt. It will also be used to form the return address of automatic notification emails sent from the system. Refer to **email return-host** under email for more information.

The **hostname** command has the following syntax:

hostname <hostname>

Related Commands

The following table summarizes other commands related to the **hostname** command:

Task	Command
Displays the system hostname. The **show hosts** output also includes details on name servers, domain name, and static host mappings.	**show hosts**
Configures a hostname for this system.	**(config) # hostname 10.10.10.10**
Deletes the hostname from this system.	**(config) # no hostname**

ib-pathway

Required Command-Line Mode = Configure

Use the ib-pathway command to configure the Resilient Inline Arrangement feature.

IMPORTANT

If you configure the Resilient Inline Arrangement feature using the GigaVUE-OS CLI, you cannot view or manage it using GigaVUE-FM. Also, if you modify this feature using the GigaVUE-OS CLI, you cannot view the changes in GigaVUE-FM. For details about how to configure this feature using GigaVUE-FM, refer to the "*Working with Flexible Inline Arrangements*" chapter in the *GigaVUE Fabric Management Guide*.

Related Commands

The following table summarizes other commands related to the **ib-pathway** command:

Task	Command
Displays the Rx and Tx statistics for either all the interbroker pathway or a specified interbroker pathway alias that are part of the inline flow deployment.	**# show ib-pathway traffic-rate {all | alias <ib_name>}**

image

Required Command-Line Mode = Enable

Required User Level = Admin

Use the **image** command to manage software images for the GigaVUE-OS® HC Series node.

The **image** command has the following syntax:

```
image
    boot <location <1 | 2> | next>
  delete <image filename>
   fetch <download URL> [filename]
   install <image filename>
   move <src filename> <dst filename>
```

The following table describes the arguments for the **image** command:

Argument	Description
boot <location <1 | 2> | next>	Specifies which of the two available boot images to boot at the next reboot. Images are installed in one of the two available boot locations—**1** or **2**. • **location**—Specifies to boot from a specified location. The **show image** command provides information on the images installed in each of the two boot locations, allowing you to select the desired image. • **next**—Specifies to boot from the next partition after the one currently booted. This argument is handy after you have installed a new image—the image is automatically installed at the location you did not boot last, so booting from the **next** location will use the image you just installed.
delete <image filename>	Deletes the specified image file. Type **image delete ?** to see a list of image filenames available for deletion. You cannot delete the currently active image file.

Argument	Description
fetch <download URL> [filename]	Retrieves the specified image file from the named location using HTTP, HTTPS, FTP, TFTP, SCP, SFTP **NOTE:** It is recommended that you use only secured protocols. Optionally, you can include a filename for the local image. The format for the download URL is as follows: **[protocol]://username[:password]@hostname/path/filename newfilename** For example, the following command uses SCP to retrieve the **ta100xx** image from the **builds** folder on 192.168.1.25: **(config) # image fetch scp://user:password@192.168.1.25/builds/ta100xx** The CLI shows you the progress of the image fetch with a series of hash marks, returning you to the system prompt when complete. For example on a GigaVUE-OS-TA100, use the **image fetch** command with the HTTP parameter to fetch the software image from an external web server as follows: **(config) # image fetch http://1.1.1.1/ta100_2016-02-17_gm.img**
install <image filename>	Installs the named image file at the specified location. The **location** argument is optional—if you do not supply it, the image is automatically installed at the next location after the one currently booted. Use **image install ?** to see a list of images available for installation. The GigaVUE-OS presents a series of status messages as it verifies and uncompresses the image, creates filesystems, and extracts the image, returning you to the system prompt when complete. For example: **(config) # image install ta100xx.img**
move <src filename> <dst filename>	Renames the specified image file. For example, the following command renames **ta100xx** as **oldimage**: **(config) # configuration moveta100xx oldimage**

Related Commands

The following table summarizes other commands related to the **image** command:

Task	Command
Displays information on available images, including currently installed images, images available for installation, the last booted partition, and the next boot partition.	**show images**
Negates image boot location changes.	**(config) # no image boot next**

inline-network

Required Command-Line Mode = Admin

Use the **inline-network** command to configure an inline network. An inline network is an arrangement of two ports of the inline-network type. The arrangement facilitates access to a bidirectional link between two networks (two far-end network devices) that need to be linked through an inline tool.

An inline network consists of inline network ports, always in pairs, running at the same speed, on the same medium (either fiber or copper). The inline network ports must be on the same GigaVUE-HC3, or GigaVUE-HC1 node.

This command is only applied to GigaVUE HC Seriesnodes. In a cluster environment, this command is only applied GigaVUE HC Series nodes through the cluster leader. The inline constructs must all be configured on one GigaVUE-HC3, or GigaVUE-HC1 node, not across nodes, even if the nodes are in a cluster.

On the GigaVUE-HC1, each TAP-HC1-G10040 module supports up to 4 inline networks. In addition, each GigaVUE-HC1 base module can support up to 8 inline networks, however, some of the ports of the base module will need to be used for inline tools, so the maximum number of inline networks on GigaVUE-HC1 is 14 with the TAP-HC1-G10040 in both modules. Starting in software version 5.0, the GigaVUE-HC1 supports a bypass combo module that supports up to 2 inline networks. Starting in software version 5.1.01, the GigaVUE-HC3 supports up to 8 inline networks.

An inline network can be unprotected or protected. Protected inline networks are implemented using bypass combo modules. Protected inline networks are based on the pairs of ports associated with physical protection switches on the bypass combo modules. The protected inline network ports provided by the bypass combo modules offer different link speeds, such as 1Gb/10Gb. Starting in software version 5.1.01, the GigaVUE-HC3 supports a 100Gb bypass combo module.

Starting in software version 4.8, the TAP-HC1-G10040 on the GigaVUE-HC1 is a copper bypass module, providing protected inline networks for copper ports.

This command is used in the inline bypass solutions described in the "*Configuring Inline Bypass Solutions*" section and in the flexible inline arrangements described in the "*Working with Flexible Inline Arrangements*" section in the *GigaVUE Fabric Management Guide*.

The **inline-network** command has the following syntax:

```
inline-network alias <alias>
  comment <comment>
  lfp enable
  hb-accept
  pair net-a <port ID or alias> and net-b <port ID or port alias>
  physical-bypass <enable | disable>
  redundancy-profile <redundancy profile alias>
  traffic-path <drop | bypass | monitoring | to-inline-tool>
```

The following table describes the arguments for the **inline-network** command.

Argument	Description
alias <alias>	Specifies the name of the inline network. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. Protected inline network aliases are created automatically on bypass combo modules. The aliases of the default inline networks are: default_inline_net_x_y_z, where x is the box ID of the node, y is the slot ID of the BPS module, and z is the port ID. Examples: **(config) # inline-network alias default_inline_net_2_3_1** **(config inline-network alias default_inline_net_2_3_1) #** **(config) # inline-network alias inNet** **(config inline-network alias inNet) #**
comment <comment>	Specifies a unique text string that describes the inline network. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config inline-network alias inNet) # comment “Inline network inNet”**
lfp enable	Specifies the link failure propagation state of either enabled or disabled. Link failure propagation controls whether the inline network link failure on one side of the inline network is propagated to the other side. The default is **enable**. When enabled, the behavior is as follows: • When the link of the side A inline network port goes up (or down), the link of the side B inline network port is brought up (or down). • When the link of the side B inline network port goes up (or down), the link of the side A inline network port is brought up (or down). For example: **(config inline-network alias inNet) # lfp enable**
hb-accept	Ensures that the inline network port pair accepts the heartbeat packets that are sent from the inline tool port pair. For details, refer to the *"Heartbeat Support Between GigaVUE Nodes"* section in the *GigaVUE-FM User's Guide*. For example: **(config inline-network alias inNet) # hb-accept**
pair net-a <port ID or alias> and net-b <port ID or port alias>	Specifies a pair of inline network ports (two ports: side A and side B). Net-a is the port identifier for the port leading to the side A network and net-b is the port identifier for the port leading to the side B network. Port identifiers can be a port ID <bid/sid/pid> or a port alias. For example: **(config inline-network alias inNet) # pair net-a 5_Net_AP1 and net-b 5_Net_BP1**
physical-bypass <enable | disable>	Controls the state of the optical protection switch on bypass combo modules or the electrical relays in copper TAP modules when the power is on. The state can be one of the following: • close—Specifies that the fiber or copper connected to the side A network port is

Argument	Description
	passively coupled with the fiber or copper connected to the side B port without any transceivers or switching fabric. Therefore, any traffic coming in is exchanged between the two inline network ports without being noticed by the system. • open—Specifies that the fiber or copper connected to the inline network ports is coupled through transceivers with the switching fabric that is under software control. Therefore, any traffic coming in is subject to the traffic forwarding rules imposed by the current configuration as well as the current state of the inline tools. When bypass combo modules or copper TAP modules are powered off, the optical protection switch or the electrical relays are always in the close state. When bypass combo modules or copper TAP modules are powered on, the state of the optical protection switch or the electrical relays are as follows: • the close state if the physical-bypass parameter is set to **enable** • the open state if the physical-bypass parameter is set to **disable** The default value of the physical-bypass parameter is **enable**. **NOTE:** The physical-bypass parameter only applies to protected inline networks. For example: **(config inline-network alias inNet) # physical-bypass disable**
redundancy-profile <redundancy profile alias>	Specifies the name of a redundancy profile for the inline network. Give the redundancy profile a name before configuring parameters under the **redundancy-profile** command. Refer to redundancy-profile. For example: **(config) # inline-network alias inNet redundancy-profile RP1**
traffic-path <drop | bypass | monitoring | to-inline-tool>	For classic inline bypass, specifies the path of the traffic received at an inline network port as follows: • **drop**—Specifies that no traffic is exchanged through the inline network ports. All traffic to these ports is dropped. No traffic is forwarded to or from the inline tool or tools. No traffic is passed from inline network port A to inline network port B or from inline network port B to inline network port A. • **bypass**—There are two cases for bypass, which take the inline maps into consideration as follows: o **If there are no inline maps associated with the inline network** or if the set of inline maps associated with the inline network guarantees that no traffic is dropped when the traffic path is set to **to-inline-tool**, then setting the traffic path to **bypass** leads to the following: all traffic arriving at the side A inline network port is forwarded to the side B inline network port and all traffic arriving at the side B inline network port is forwarded to the side A inline network port through a logical bypass. o **If the set of inline maps associated with the inline network involves some traffic drop** when the traffic path is set to **to-inline-tool**, then setting the traffic path to **bypass** leads to the following: all traffic arriving at the side A inline network port that would *not* have been dropped with traffic path set to **to-inline-tool** is forwarded to the side B inline network port and all traffic arriving at the side B inline network port that would *not* have been dropped with traffic path set to **to-inline-tool** is forwarded to the side A inline network port through a logical bypass. o In either of these bypass cases, no traffic is forwarded to the inline tool or tools. • **monitoring**—There are two cases for monitoring, which take the inline maps into

Argument	Description
	consideration as follows: o **If there are no inline maps associated with the inline network** or if the set of inline maps associated with the inline network guarantees that no traffic is dropped when the traffic path is set to **to-inline-tool**, then setting the traffic path to **monitoring** leads to the following: all traffic is forwarded as for **bypass**, but a copy of the traffic is forwarded to the inline tool or tools according to the configured maps between the inline network and the inline tool or tools. o **If the set of inline maps associated with the inline network involves some traffic drop** when the traffic path is set to **to-inline-tool**, then setting the traffic path to **monitoring** leads to the following: all traffic that would *not* have been dropped with traffic path set to **to-inline-tool** is forwarded as for **bypass**, but a copy of the traffic is forwarded to the inline tool or tools according to the configured maps between the inline network and the inline tool or tools. o In either of these monitoring cases, no traffic is taken from the inline tools. • **to-inline-tool**—Specifies that traffic received at the inline network ports is forwarded according to the following factors: o the configured inline maps between the inline network and the inline tools o the failover actions of the inline tool or tools o the health state of the inline tool or tools o The default is **bypass**. For example: **(config inline-network alias inNet) # traffic-path to-inline-tool** **NOTE**: • Prior to software version 4.4, the default was **drop**. • If the inline network is not associated with an inline tool or an inline tool group through a map or map passall, **monitoring** is the same as **bypass**. • If the **traffic-path** is set to **monitoring**, when a map is configured for the inline network, the monitoring map will be established immediately. • If the inline network **traffic-path** is set to **bypass**, and there is a rule-based map from the inline network to a destination, the map's counters are incremented for traffic that matches the rule. However, the traffic is actually being safely bypassed to other side of the network and these counters can be ignored. It is also recommended to use a collector to avoid packet drops on unmatched traffic.
traffic-path <drop | bypass | monitoring | to-inline-tool>	For flexible inline arrangements, the descriptions of **traffic-path** are as follows: • **to-inline-tool**—all traffic originating from the inline network is directed to the sequence of inline tools and inline tool groups and is guided through the inline tools and inline tool groups according to the current inline tool and inline tool group status. • **bypass**—all traffic that, with the traffic path parameter set to **to-inline-tool** would go toward the configured sequence of inline tools and inline tool groups, is re-directed to the opposite-side inline network port. • **drop**—all traffic originating from the inline network is dropped. • **monitoring**—a copy of the traffic originating from the inline network is handled just as for the traffic path of **bypass** and another copy is handled just as for the traffic path of **to-inline-tool**, except that no traffic of the second copy is sent to the exit port. • The default is **to-inline-tool**.

Argument	Description
	With the traffic path **monitoring**, for each sequence originating from the inline network, the system guides two copies of the traffic but only the bypass copy reaches the opposite inline network port. The other copy visits all the inline tools and inline tool groups in the sequence just as if the inline network was set to **to-inline-tool**, but the supply of traffic from the inline tool returning any traffic is redirected to a null VLAN (with no member ports).

Related Commands

The following table summarizes other commands related to the **inline-network** command:

Task	Command
Displays a specified inline network.	**# show inline-network alias inNet**
Displays all inline networks.	**# show inline-network all**
Displays all inline networks in table format.	**# show inline-network brief**
Displays Forwarding State, which is the current status of the inline bypass solution. The heartbeat statistics for the inline network are also displayed.	**# show inline-network**
Displays the Rx and Tx statistics for all the inline networks that are part of the inline flow deployment.	**# show inline-network traffic-rate all**
Displays the Rx and Tx statistics for the specified inline network alias that is part of the inline flow deployment.	**# show inline-network traffic-rate alias <alias_ name>**
Deletes a specified inline network. **Note**: If the inline network is protected, this command is invalid because you cannot delete a protected inline network.	**(config) # no inline-network alias inNet**
Deletes the comment for this inline network.	**(config) # no inline-network alias inNet comment**
Disables the link failure propagation state.	**(config) # no inline-network alias inNet lfp enable**
Disables the physical bypass state.	**(config) # no inline-network alias inNet physical-bypass enable**
Disables a redundancy profile.	**(config) # no inline-network alias inNet redundancy-profile**
Deletes all inline networks.	**(config) # no inline-network all**

Task	Command
Clears all the heartbeat statistics for the specified inline network.	**(config) # clear hb-counters inline-net alias inNet**
Clears the heartbeat statistics for all the inline networks that are part of the flexible inline flow deployment.	**(config) # clear hb-counters inline-net all**
Configures the port speed for an inline network. This command only applies to protected ports. **Note**: When you configure the port speed on one of the two inline network ports of a given inline network, the other inline network port is automatically configured to the same speed.	**(config) # port 4/1/x17 params speed 1000**

inline-network-group

Required Command-Line Mode = Admin

Use the **inline-network-group** command to configure an inline network group. An inline network group is an arrangement of multiple inline networks that share the same inline tool or tools. Use this command to specify the list of inline networks in the group.

The inline network ports that make up the inline networks participating in the inline network group are always in pairs, running at the same speed, on the same medium (fiber or copper). All inline network ports of the inline network group must be on the same GigaVUE-HC3, or GigaVUE-HC1 node. The inline networks participating in the inline network group can be different speeds and different mediums.

This command is only applied to GigaVUE HC Series nodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader. The inline constructs must all be configured on one GigaVUE-HC3, or GigaVUE-HC1 node, not across nodes, even if the nodes are in a cluster.

This command is used in the inline bypass solutions described in the "*Configuring Inline Bypass Solutions"* section in the *GigaVUE Fabric Management Guide*.

The **inline-network-group** command has the following syntax:

```
inline-network-group [alias <alias>]
  comment <comment>
  network-list <inline-network list>
```

The following table describes the arguments for the **inline-network-group** command.

Argument	Description
alias <alias>	Specifies the name of the inline network group. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) #**
comment <comment>	Specifies a unique text string that describes the inline network group. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config inline-network-group alias inNetGroup) # comment "Inline network group inNetGroup"**
network-list <inline network list>	Specifies the list of aliases of inline networks participating in sharing the same inline tool or tools. Each GigaVUE HC Series node supports up to 12 inline networks in a single network-list. The inline networks in the network-list can be different speeds, such as 10Gb and 1Gb, and different mediums (fiber and copper). Separate each alias with a comma. For example: **(config inline-network-group alias inNetGroup) # network-list in11,in12,default_inline_net_3_2_1,default_inline_net_3_2_2**

Related Commands

The following table summarizes other commands related to the **inline-network-group** command:

Task	Command
Displays inline network groups.	**# show inline-network-group**
Displays a specified inline network group.	**# show inline-network-group alias inNetGroup**
Displays all inline network groups.	**# show inline-network-group all**
Displays the Rx and Tx statistics for all the inline network groups that are part of the inline flow deployment.	**# show inline-network-group traffic-rate all**
Displays the Rx and Tx statistics for the specified inline network group alias that is part of the inline flow deployment.	**# show inline-network-group traffic-rate alias <alias_name>**
Deletes the comment for this inline network group.	**(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) # no comment**

Task	Command
Deletes a specified inline network group.	**(config) # no inline-network-group alias inNetGroup**
Deletes the network list for the specified inline network group.	**(config) # no inline-network-group alias inNetGroup network-list**
Deletes all inline network groups.	**(config) # no inline-network-group all**

inline-serial

Required Command-Line Mode = Admin

Use the **inline-serial** command to configure inline tools or inline tool groups in a series, in which the traffic from one side of the inline network is guided through the members of the series before it is sent out the other side of the inline network. With inline tools and inline tool groups arranged in a series, the traffic from one inline tool or inline tool group flows to the next, so all tools in the series see the same traffic.

The inline tool ports that make up the inline tools and inline tool groups participating in the series are always in pairs, running at the same speed, on the same medium (fiber or copper). All inline tool and inline tool group ports of the series must be on the same GigaVUE-HC3, or GigaVUE-HC1 node. The inline tool and inline tool group ports must also be on the same GigaVUE-HC3, or GigaVUE-HC1 node as the inline network ports.

This command is only applied to GigaVUE HC Series nodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader. The inline constructs must all be configured on one GigaVUE-HC3, or GigaVUE-HC1 node, not across nodes, even if the nodes are in a cluster.

This command is used in the inline bypass solutions described in the *"Configuring Inline Bypass Solutions"* section in the *GigaVUE Fabric Management Guide*.

The **inline-serial** command has the following syntax:

```
inline-serial alias <alias>
  comment <comment>
  enable
  failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-
forced-down |      per-tool>
  inline-tool-list <list of inline tools and inline tool groups>
  per-direction-order <reverse | forward>
```

The following table describes the arguments for the **inline-serial** command.

Argument	Description
alias <alias>	Specifies the name of the inline tool series. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # inline-serial alias inSer** **(config inline-serial alias inSer) #**
comment <comment>	Specifies a unique text string that describes the inline tool series. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config inline-serial alias inSer) # comment “Inline serial inSer”**
enable	Enables or disables the inline tool series. Use **enable** to put the inline tool series into service. Disable the inline tool series to simulate a failure or to take the inline tool series offline for maintenance purposes. The default is disabled. For example, to enable: **(config inline-serial alias inSer) # enable** For example, to disable: **(config inline-serial alias inSer) # no enable**
failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-forced-down | per-tool>	Specifies the failover action taken in response to a failure of an inline tool series as a whole. An inline tool series is declared to be in a failure condition as soon as any of its member inline tools goes into a failure condition. An inline tool series recovers from a failure condition after all the member inline tools recover from their failure conditions. The failover-action attributes of the individual inline tools participating in an inline tool series are ignored. Instead, the failover-action configured for the inline tool series is respected. The values are as follows: • **tool-bypass**—Specifies that when the inline tool series fails, the traffic that normally was directed to the inline tool is redirected to the bypass path. Use this failover action for configurations involving an inline tool series that is associated with an inline network using rule-based maps. For configurations using map passalls, **tool-bypass** is the same as **network-bypass**. • **tool-drop**—Specifies that when the inline tool series fails, the, the traffic that normally was directed to the inline tool is dropped. Use this failover action for configurations involving an inline tool series that is associated with an inline network using rule-based maps. For configurations using map passalls, **tool-drop** is the same as **network-drop**. • **network-bypass**—Specifies that when the inline tool series fails, all traffic that would not have been dropped when the inline network or networks had a NORMAL forwarding state is directed to the bypass path. That is, all such traffic arriving at the side A inline network port or ports is forwarded to the side B inline network port or ports and all traffic arriving at the side B inline network port or ports is forwarded to the side A inline network port or ports. • **network-drop**—Specifies that when the inline tool series fails, all traffic coming to the respective inline network (or inline network group) is dropped.

Argument	Description
	• **network-port-forced-down**—Specifies that when the inline tool series fails, the inline network ports of the respective inline network are forced down. • **per-tool**—Specifies that when an individual inline tool in the series fails, the action depends on the failover action of the individual inline tool, as configured with the **inline-tool failover-action** command. Each inline tool in a series can have its own failover action. For example, if the failover action of the inline series is configured as **per-tool** and the failover action of an individual inline tool in the series is configured as **tool-bypass**, when that tool in the series fails, the traffic will skip over the failed tool. **Note**: The **per-tool failover-action** is not supported for SSL Decryption for inline tools. • The default is **tool-bypass**. For example: **(config inline-serial alias inSer) # failover-action tool-drop** **Note**: Before changing the failover action, enable the inline tool series using the **enable** command. Refer to the "*Inline Tool Series Global Failover Action*" and "*Inline Tool Series Local Failover Action*" in the *GigaVUE Fabric Management Guide* for details.
<list of inline tools and inline tool groups>	Specifies the list of aliases of inline tools and inline tool groups participating in the inline tool series. **Note**: Specify the inline tools and inline tool groups in the inline tool list in the order of your configuration. To change the order, override the existing list with a new one. Separate each alias with a comma. For example: **(config inline-serial alias inSer) # inline-tool-list IT_001,ITG_002,IT_003,ITG_004** The number of inline tools and inline tool groups in the inline series is limited only by the number of ports available for creating the inline networks and inline tools participating in the inline bypass solution. When an inline tool group is included as a member of an inline series, the following applies: • a spare inline tool can be configured on the inline tool group • inline maps to individual members of an inline tool group are not supported • asymmetrical hashing is not supported, which means that the hash options, a-srcip-b-dstip and b-srcip-a-dstip, are not allowed on the inline tool group
per-direction-order <reverse | forward>	Specifies the per-direction order of the side B traffic of the inline tool series. This parameter configures the traffic direction order of side B traffic with respect to the inline tool list order as follows: • **reverse**—Specifies that the traffic from network B is to flow through the inline tool list in reverse order, for example, from the third tool, to the second tool, to the first tool. • **forward**—Specifies that the traffic from network B is to flow through the inline tool list in the order it which it is defined, for example, from the first tool, to the second tool, to the third tool. • The default is **reverse**. For example:

Argument	Description
	(config inline-serial alias inSer) # per-direction-order forward **Note**: Traffic from network A to network B for both **reverse** and **forward** flows from the first tool, to the second tool, to the third tool. Refer to the *"Inline Tool Series Per-Direction Order"* section in the *GigaVUE Fabric Management Guide* for details.

Related Commands

The following table summarizes other commands related to the **inline-serial** command:

Task	Command
Displays inline tool series.	**# show inline-serial**
Displays a specified inline tool series.	**# show inline-serial alias inSer**
Displays all inline tool series.	**# show inline-serial all**
Displays the Rx and Tx statistics for either all the inline serials or a specified inline serial alias that are part of the inline flow deployment.	**# show inline-serial traffic-rate {all | alias <serial_name>}**
Deletes a specified inline tool series.	**(config) # no inline-serial alias inSer**
Deletes the comment for this inline tool series.	**(config) # no inline-serial alias inSer comment**
Disables a specified inline tool series.	**(config) # no inline-serial alias inSer enable**
Deletes the inline tool list for the specified inline series.	**(config) # no inline-serial alias inSer inline-tool-list**
Deletes all inline tool series.	**(config) # no inline-serial all**

inline-tool

Required Command-Line Mode = Admin

There are two meanings to the term inline tool. The inline tool software construct consists of a pair of inline tool ports plus the inline tool attached to the ports. The software construct has attributes that are configured on the GigaVUE-HC3, and GigaVUE-HC1 nodes.

The term inline tool also refers to the pass-through device itself that performs packet inspection and selective forwarding, such as an Intrusion Protection System (IPS). This is a physical device, external to the GigaVUE HC Seriesnode.

Use the **inline-tool** command to configure the inline tool software construct. An inline tool consists of inline tool ports, always in pairs, running at the same speed, on the same medium (fiber or copper). The inline tool ports must be on the same GigaVUE-HC3, or GigaVUE-HC1 node. The inline tool ports must also be on the same GigaVUE-HC3, or GigaVUE-HC1 node as the inline network ports.

This command is only applied to GigaVUE HC Series nodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader. The inline constructs must all be configured on one GigaVUE-HC3, or GigaVUE-HC1 node, not across nodes, even if the nodes are in a cluster.

Each GigaVUE-HC3 supports up to 48 inline tools.

On the GigaVUE-HC1, the base module can be used for inline tools. It supports up to 8 inline tools. On the GigaVUE-HC1, the bypass combo module can support up to 4 inline tools.

This command is used in the inline bypass solutions described in the *"Configuring Inline Bypass Solutions"* section and in the flexible inline arrangements described in the "*Working with Flexible Inline Arrangements"* section in the *GigaVUE Fabric Management Guide*.

The **inline-tool** command has the following syntax:

```
inline-tool alias <alias>
  comment <comment>
  enable
  inline-tool-type <external | gmon>
  failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-
forced-down>
  flex-traffic-path <to-inline-tool | bypass | monitoring | drop>
  hb-ip-addr-a <tool-a heartbeat IP address>
  hb-ip-addr-b <tool-b heartbeat IP address>
  hb-profile <hb-profile alias | default>
  heart-beat
  negative-heart-beat
  nhb-profile <negative heartbeat profile alias>
  pair tool-a <port ID or port alias> and tool-b <port ID or port alias>
  recover
  recovery mode <automatic | manual>
  shared <true | false>
```

The following table describes the arguments for the **inline-tool** command.

Argument	Description
alias <alias>	Specifies the name of the inline tool. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # inline-tool alias inTool** **(config inline-tool alias inTool) #**
comment <comment>	Specifies a unique text string that describes the inline tool. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config inline-tool alias inTool) # comment "Inline Tool inTool"**
enable	Enables or disables the inline tool. Use **enable** to put the inline tool into service. Use

Argument	Description
	disable to simulate an inline tool failure or to take the inline tool offline for maintenance purposes. The default is disabled. For example, to enable the inline tool: **(config inline-tool alias inTool) # enable** For example, to disable the inline tool: **(config inline-tool alias inTool) # no enable**
inline-tool-type	Configures the inline tool type as follows: • **external**—To configure a third-party tool. • **gmon**—To configure a GigaVUE node as a tool. The minimum timeout for heartbeat sessions for the inline tool of type, **gmon** is 200 milliseconds. Heartbeat profiles configured with timeout less than 200 milliseconds cannot be attached to the inline tool of type, **gmon**. Negative heartbeat profiles cannot be attached with the inline tool of type, **gmon**. For details, refer to the *"Heartbeat Support Between GigaVUE Nodes"* section in the *GigaVUE-FM User's Guide*. The default is **external**. For example, to configure a third-party tool: **(config inline-tool alias inTool) # inline-tool-type external** For example, to configure a GigaVUE node as a tool: **(config inline-tool alias inTool) # inline-tool-type gmon**
failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-forced-down>	Specifies the failover action taken in response to a failure of an inline tool as follows: • **tool-bypass**—Specifies that when the inline tool fails, the traffic that normally was directed to the inline tool is redirected to the bypass path. Use this failover action for configurations involving multiple inline tools associated with an inline network or inline network group using rule-based maps. For configurations using map passalls, **tool-bypass** is the same as **network-bypass**. • **tool-drop**—Specifies that when the inline tool fails, the traffic that normally was directed to the inline tool is dropped. Use this failover action for configurations involving multiple inline tools associated with an inline network or inline network group using rule-based maps. For configurations using map passalls, **tool-drop** is the same as **network-drop**. • **network-bypass**—Specifies that when the inline tool fails, all traffic that would not have been dropped when the inline network or networks had a NORMAL forwarding state is directed to the bypass path. That is, all such traffic arriving at the side A inline network port or ports is forwarded to the side B inline network port or ports and all traffic arriving at the side B inline network port or ports is forwarded to the side A inline network port or ports. • **network-drop**—Specifies that when the inline tool fails, all traffic coming to the respective inline network (or inline network group) is dropped. • **network-port-forced-down**—Specifies that when the inline tool fails, the inline network ports of the respective inline network (or inline network group) are forced down. • The default is **tool-bypass**. For example: **(config inline-tool alias inTool) # failover-action tool-drop**

Argument	Description
	Note: Before changing the failover action, enable the inline tool using the **enable** command.
flex-traffic-path <to-inline-tool | bypass | monitoring | drop>	For flexible inline arrangements, provides per-tool traffic path options. Each inline tool or inline tool group involved in a flexible inline map can specify its own traffic path. The options are as follows, however the behavior of the traffic will depend on a variety of factors including the inline tools in the sequence, their individual **flex-traffic-path** settings, the operational state of the inline tools, and the direction of traffic: • **to-inline-tool**—traffic is forwarded from the inline tool. • **bypass**—traffic bypasses the inline tool. Use this option for performing maintenance on an inline tool. • **drop**—traffic is dropped at the inline tool. • **monitoring**—traffic is fed to the inline tool and absorbed, while a copy of the traffic is sent to the next inline tool in the sequence. Traffic returned from side B of the network is also absorbed at the inline tool in monitoring mode. • The default is **to-inline-tool**. For example: **(config inline-tool alias inTool) # flex-traffic-path drop**
hb-ip-addr-a <tool-a heartbeat IP address>	Specifies heartbeat IP address A, which is the destination IP address to be used in heartbeat packets sent from side A to side B. The default is N.N.N.N, where N is the port number within the chassis as shown on the face plate. This parameter applies only to heartbeat profiles that use a standard ICMP ARP packet. For example: **(config inline-tool alias inTool) # hb-ip-addr-a 1.1.1.1**
hb-ip-addr-b <tool-b heartbeat IP address>	Specifies heartbeat IP address B, which is the destination IP address to be used in heartbeat packets sent from side B to side A. The default is N.N.N.N, where N is the port number within the chassis as shown on the face plate. This parameter applies only to heartbeat profiles that use a standard ICMP ARP packet. For example: **(config inline-tool alias inTool) # hb-ip-addr-b 2.2.2.2**
hb-profile <hb-profile alias | default>	Specifies the name of a heartbeat profile containing the heartbeat parameters to be used if the heartbeat mechanism is enabled for this inline tool. The default heartbeat profile alias is **default**. For example, to specify the heartbeat profile to associate with this inline tool: **(config inline-tool alias inTool) # hb-profile hb_5** or **(config inline-tool alias inTool) # hb-profile default** For example, to delete the heartbeat profile associated with this inline tool: **(config inline-tool alias inTool) # no hb-profile hb_5** Refer to hb-profile.
heart-beat	Specifies the state of the heartbeat as enabled or disabled. When enabled, this parameter controls the use of the heartbeat mechanism for the specified inline tool. The default is **disabled**. For example, to enable the heartbeat:

Argument	Description
	(config inline-tool alias inTool) # heart-beat For example, to disable the heartbeat: **(config inline-tool alias inTool) # no heart-beat**
negative-heart-beat	Specifies the state of the negative heartbeat as enabled or disabled. When enabled, this parameter controls the use of the negative heartbeat mechanism for the specified inline tool. The default is **disabled**. For example, to enable the negative heartbeat: **(config inline-tool alias inTool) # negative-heart-beat** For example, to disable the negative heartbeat: **(config inline-tool alias inTool) # no negative-heart-beat**
nhb-profile <negative heartbeat profile alias>	Specifies the name of a negative heartbeat profile containing the heartbeat parameters to be used if the negative heartbeat mechanism is enabled for this inline tool. For example to specify the negative heartbeat profile to associate with this inline tool: **(config inline-tool alias inTool) # nhb-profile nhb_1** For example, to delete the negative heartbeat profile associated with this inline tool: **(config inline-tool alias inTool) # no nhb-profile nhb_1** Refer to nhb-profile.
pair tool-a <port ID or port alias> and tool-b <port ID or port alias>	Specifies a pair of inline tool ports (two ports: side A and side B). Tool-a is the port identifier for the port leading to the side A inline tool and tool-b is the port identifier for the port leading to the side B inline tool. Port identifiers can be a port ID <bid/sid/pid> or a port alias. For example: **(config inline-tool alias inTool) # pair tool-a iT1 and tool-b iT2**

Argument	Description
recover	Puts an inline tool back into service if the recovery mode is configured as **manual** and the inline tool has an operational state of ready. For example: **(config inline-tool alias inTool) # recover**
recovery mode <automatic | manual>	Configures the recovery mode for each inline tool. After an inline tool goes down, the following modes specify how to bring it back up after it has recovered: • **automatic**—Specifies automatic recovery, which redirects traffic back to the inline tool as soon as it has recovered from all faulty conditions. • **manual**—Specifies manual recovery, which lets you control when to put an inline tool back into service after the tool has recovered using a CLI command. For example, you can wait for a maintenance window to return the inline tool to service. • The default is **automatic**. For example: **(config inline-tool alias inTool) # recovery mode manual** If the recovery mode is **manual**, use the **recover** command to put the inline tool back into service. Refer to the "*Inline Tool Recovery Mode*" section in the *GigaVUE Fabric Management Guide* for details.
shared <true | false>	Specifies how an inline tool is going to be shared as follows: • **true**—Specifies that the inline tool is going to be shared by different sources. • **false**—Specifies that the inline tool will not be shared by different sources. • The default is **false**. When **shared** is enabled (true), the inline tool can receive traffic from multiple sources (inline networks). This means that the inline tool can be used in a map in which the source is an inline network group. The **shared** parameter is also used for SSL Decryption for inline tools, when the source is GigaSMART. For an inline network group, shared must be **true** because traffic is received from multiple sources. An inline tool group or inline series does not have its own shared setting. The shared setting is derived from the inline tools. Therefore all the members in an inline tool group or inline series must have the same setting. For example, if an inline tool group has three inline tool members, the shared setting of all three inline tools must be the same. When an inline tool is shared (true), the decrypted traffic will be VLAN tagged. The connected inline device is expected to receive VLAN tagged packets instead of untagged packets. There is an extra outer VLAN tag added to the packet, which the connected inline device needs to see. When an inline tool is not shared (false), the extra VLAN tag is not added. This allows untagged traffic to be sent to the tool ports. Starting in software release 5.2 for SSL Decryption for inline tools, **false** is supported for inline tools that are not able to handle more than one VLAN tag, such as Q-in-Q tagged packets. Thus, an inline SSL map can be configured from an inline network or inline network group to an inline tool, inline tool group, or inline series. When an inline tool is not shared (false), the inline tool can be used in only one flexible

Argument	Description
	inline map. For example: **(config inline-tool alias inTool) # shared true**

Related Commands

The following table summarizes other commands related to the **inline-tool** command:

Task	Command
Displays inline tools, which displays the status of the inline tool ports and the heartbeat.	**# show inline-tool**
Displays a specified inline tool.	**# show inline-tool alias inTool**
Displays all inline tools.	**# show inline-tool all**
Displays all inline tools in brief format.	**# show inline-tool brief**
Displays the Gigamon VLAN IDs for all inline tools.	**# show inline-tool vlan-mapping**
Displays the Rx and Tx statistics for all the inline tools that are part of the inline flow deployment.	**# show inline-tool traffic-rate all**
Displays the Rx and Tx statistics for the specified inline tool alias that is part of the inline flow deployment.	**# show inline-tool traffic-rate alias <alias_name>**
Deletes a specified inline tool.	**(config) # no inline-tool alias inTool**
Deletes the comment for this inline tool.	**(config) # no inline-tool alias inTool comment**
Disables an inline tool.	**(config) # no inline-tool alias inTool enable**
Deletes the heartbeat IP address associated with inline tool a.	**(config) # no inline-tool alias inTool hb-ip-addr-a**
Deletes the heartbeat IP address associated with inline tool b.	**(config) # no inline-tool alias inTool hb-ip-addr-b**
Deletes the heartbeat profile associated with this inline tool.	**(config) # no inline-tool alias inTool hb-profile**
Disables the heartbeat associated with this inline tool.	**(config) # no inline-tool alias inTool heart-beat**
Disables the negative heartbeat associated with this inline tool.	**(config) # no inline-tool alias inTool negative-heart-beat**
Deletes the negative heartbeat profile associated with this inline tool.	**(config) # no inline-tool alias inTool nhb-profile**
Deletes the tool port list of this inline tool.	**(config) # no inline-tool alias inTool pair**

Task	Command
Deletes all inline tools.	**(config) # no inline-tool all**
Clears all the heartbeat statistics for the specified inline tool.	**(config) # clear hb-counters inline-tool alias inTool**
Clears the heartbeat statistics for all the inline tools that are part of the flexible inline flow deployment.	**(config) # clear hb-counters inline-tool all**

inline-tool-group

Required Command-Line Mode = Admin

Use the **inline-tool-group** command to configure an inline tool group. An inline tool group is an arrangement of multiple inline tools to which traffic is distributed to the inline tools based on hardware-calculated hash values. For example, if one tool goes down, traffic is redistributed to other tools in the group using hashing.

You also use the **inline-tool-group** command to configure redundancy, such as 1+1 and N+1.

The inline tool ports that make up the inline tools participating in the inline tool group are always in pairs, running at the same speed, on the same medium (fiber or copper). All inline tool ports of the inline tool group must be on the same GigaVUE-HC3 node, but can be on different modules on the node. On the GigaVUE-HC1, all the inline tool ports of the inline group must be on either the base module or the bypass combo module. The inline tool ports must also be on the same GigaVUE-HC3, or GigaVUE-HC1 node as the inline network ports.

This command is only applied to GigaVUE HC Series nodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader. The inline constructs must all be configured on one GigaVUE-HC3, or GigaVUE-HC1 node, not across nodes, even if the nodes are in a cluster.

This command is used in the inline bypass solutions described in the *"Configuring Inline Bypass Solutions"* section and in the flexible inline arrangements described in the "*Working with Flexible Inline Arrangements"* section in the *GigaVUE Fabric Management Guide*.

The **inline-tool-group** command has the following syntax:

```
inline-tool-group alias <alias>
  comment <comment>
  enable
  failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-
forced-down>
  failover-mode spread
  flex-traffic-path <to-inline-tool | bypass | monitoring | drop>
  hash <advanced | a-srcip-b-dstip | b-srcip-a-dstip>
  minimum-group-healthy-size <number>
  release-spare-if-possible
```

spare-inline-tool <spare inline tool alias>
tool-list <inline-tool list>
hash-weights <inline-tool weights>

The following table describes the arguments for the **inline-tool-group** command.

Argument	Description
alias <alias>	Specifies the name of the inline tool group. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) #**
comment <comment>	Specifies a unique text string that describes the inline tool group. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config inline-tool-group alias inToolGroup) # comment "Inline tool group inToolGroup"**
enable	Enables or disables the inline tool group. Use enable to put the inline tool group into service. Disable the inline tool group to simulate a failure or to take the inline tool group offline for maintenance purposes. The default is disabled. For example, to enable: **(config inline-tool-group alias inToolGroup) # enable** For example, to disable: **(config inline-tool-group alias inToolGroup) # no enable**
failover-action <tool-bypass | tool-drop | network-bypass | network-drop | network-port-forced-down>	Specifies the failover action taken in response to a failure of an inline tool group, when the number of healthy inline tools in the inline tool group (including the spare inline tool, if configured) falls below the configured minimum. The values are as follows: • **tool-bypass**—Specifies that when the inline tool group fails, the traffic that normally was directed to the inline tool group is redirected to the bypass path. Use this failover action for configurations involving multiple inline tools or inline tool groups associated with an inline network or inline network group using rule-based maps. For configurations using map passalls, **tool-bypass** is the same as **network-bypass**. • **tool-drop**—Specifies that when the inline tool group fails, the traffic that normally was directed to the inline tool group is dropped. Use this failover action for configurations involving multiple inline tools or inline tool groups associated with an inline network or inline network group using rule-based maps. For configurations using map passalls, **tool-drop** is the same as **network-drop**. • **network-bypass**—Specifies that when the inline tool group fails, all traffic that would not have been dropped when the inline network or networks had a NORMAL forwarding state is directed to the bypass path. That is, all such traffic arriving at the side A inline network port or ports is forwarded to the side B inline network port or ports and all traffic arriving at the side B inline network port or ports is forwarded to the side A inline network port or ports. • **network-drop**—Specifies that when the inline tool group fails, all traffic coming to the

Argument	Description
	respective inline network (or inline network group) is dropped. • **network-port-forced-down**—Specifies that when the inline tool group fails, the inline network ports of the respective inline network (or inline network group) are forced down. The default is **tool-bypass**. The failover actions of the individual inline tools belonging to an inline tool group are ignored when the individual inline tools encounter failure conditions. Instead, the failover mechanism of the inline tool group is used to distribute traffic among the set of healthy inline tools and declare the group to be in a failure condition when the number of healthy inline tools falls below a configured minimum. For example: **(config inline-tool-group alias inToolGroup) # failover-action tool-drop** **NOTE:** Before changing the failover action, enable the inline tool group using the **enable** command.
failover-mode spread	Specifies the failover mode of the inline tool group. The mode specifies how to handle a failure of an individual member of the inline tool list when a spare inline tool is either not configured or has failed. The only value is spread, which redistributes all the traffic coming from the inline network (or inline network group) to the reduced set of inline tools, excluding the failed inline tool or tools. **NOTE:** If there is only one inline tool in the tool list, this parameter is not used. For example: **(config inline-tool-group alias inToolGroup) # failover-mode spread**
flex-traffic-path <to-inline-tool | bypass | monitoring | drop>	For flexible inline arrangements, provides per-tool traffic path options. Each inline tool or inline tool group involved in a flexible inline map can specify its own traffic path. The options are as follows, however the behavior of the traffic will depend on a variety of factors including the inline tools in the sequence, their individual **flex-traffic-path** settings, the operational state of the inline tools, and the direction of traffic: • **to-inline-tool**—traffic is forwarded from the inline tool. • **bypass**—traffic bypasses the inline tool. Use this option for performing maintenance on an inline tool. • **drop**—traffic is dropped at the inline tool. • **monitoring**—traffic is fed to the inline tool and absorbed, while a copy of the traffic is sent to the next inline tool in the sequence. Traffic returned from side B of the network is also absorbed at the inline tool in monitoring mode. The default is **to-inline-tool**. For example: **(config inline-tool-group alias inToolGroup) # flex-traffic-path drop**
hash <advanced | a-srcip-b-dstip | b-srcip-a-dstip>	Specifies the type of hashing for distributing packets across a number of inline tools belonging to an inline tool group. The values are as follows: • **advanced**—Specifies symmetrical hashing, which is derived from the combination of packet fields based on the criteria selected for the advanced-hash algorithm by using the **gigastream advanced-hash** command. For inline bypass applications, the most common choice of criteria for the advanced-hash algorithm is the combination of

Argument	Description
	source IP and destination IP addresses. This produces a hash value that sends all traffic associated with the same session to the same inline tool in the inline tool group. • **a-srcip-b-dstip**—Specifies asymmetrical hashing, which is derived from the source IP address for side A of the network and the destination IP address for side B of the network. This produces a hash value that sends all traffic associated with the same source address residing on side A to the same inline tool in the inline tool group, regardless of destination or session. • **b-srcip-a-dstip**—Specifies asymmetrical hashing, which is derived from the destination IP address for side A of the network and the source IP address for side B of the network. This produces a hash value that sends all traffic associated with the same source address residing on side B to the same inline tool in the inline tool group, regardless of destination or session. The default is **advanced**. For flexible inline arrangements, asymmetrical hashing (using a-srcip-b-dstip and b-srcip-a-dstip) is not supported. **NOTE:** If there is only one inline tool in the tool list, the **hash** parameter is not used. Examples: **(config inline-tool-group alias inToolGroup) # hash a-srcip-b-dstip** **(config inline-tool-group alias inToolGroup) # hash advanced** Refer to the *"Symmetrical and Asymmetrical Hashing"* section in the GigaVUE Fabric Management Guide for details. Restrictions are included under "*Asymmetrical Hashing Restrictions" section in the GigaVUE Fabric Management Guide*.
minimum-group-healthy-size <number>	Specifies the minimum number of inline tools that must be *up* so that the entire inline tool group is considered to be *up*. The minimum number includes the inline tools in the inline tool list plus the spare inline tool, if one is configured. The range is 1 to 16. The default is 1. **NOTE:** If there is only one inline tool in the tool list, this parameter is not used. For example: **(config inline-tool-group alias inToolGroup) # minimum-group-healthy-size 3**
release-spare-if-possible	Determines the behavior of the spare inline tool after it has replaced an original inline tool in the inline tool group following an inline tool failure. The values are as follows: • **Disabled**—Specifies that the spare inline tool will remain active regardless of the health of the originally failed inline tool. Even if the original inline tool recovers, the spare that replaced it will remain in the active set of inline tools. • **Enabled**—Specifies that after the original inline tool recovers, the spare that replaced it will be released, if possible, from the active set of tools to become the spare again. The default is disabled. **NOTE:** If a spare inline tool is not configured, this parameter is not used. For example: **(config inline-tool-group alias inToolGroup) # release-spare-if-possible**
spare-inline-tool	Specifies the alias of an inline tool to which traffic is forwarded when the first failure

Argument	Description
<spare inline tool alias>	occurs in the set of primary inline tools. The default is blank (not specified). If a spare inline tool is configured, the inline tool group becomes a redundant arrangement of inline tools. For example: **(config inline-tool-group alias inToolGroup) # spare-inline-tool IT_004**
tool-list <inline-tool list>	When a spare inline tool is not configured, specifies the list of inline tool aliases participating in hash-based traffic distribution. The number of inline tool aliases in the list is between 1 and 16. When a spare inline tool is configured, specifies the list of aliases of primary inline tools to which traffic is forwarded as long as they are all healthy. The number of inline tools in the list is between 2 and 16. Separate each inline tool alias in the list with a comma. For example: **(config inline-tool-group alias inToolGroup) # tool-list IT_001,IT_002,IT_003**
hash-weights <inline-tool weights>	Specifies the relative weight or default weight for the inline tools based on which traffic is directed to the inline tools in the inline tool group. If you assign a default weight, the traffic is distributed equally to all the inline tools in a group. If an inline tool in a group goes down and the group maintains the **minimum-group-healthy-size** defined for the group, the traffic is redistributed to the remaining inline tools based on the default weight or the relative weight assigned to the inline tools. If the inline tool group does not meet the **minimum-group-healthy-size** defined for the group, the traffic is redistributed based on the **failover-action** defined for the group. The valid range is 1–256. **NOTE:** Resilient hashing is not supported for classic inline maps. You cannot add a spare-inline-tool, if you have assigned hash weights to the inline tools in a group. Separate each inline tool weight in the list with a comma. For example: • Relative weights: **(config inline-tool-group alias inToolGroup) # tool-list IT_001,IT_002,IT_003 hash-weights 2,3,2** • Default weight: **(config inline-tool-group alias inToolGroup) # tool-list IT_001,IT_002,IT_003 hash-weights default**

Related Commands

The following table summarizes other commands related to the **inline-tool-group** command:

Task	Command
Displays inline tool groups.	**# show inline-tool-group**
Displays a specified inline tool group.	**# show inline-tool-group alias inToolGroup**
Displays all inline tool groups.	**# show inline-tool-group all**
Displays the Rx and Tx statistics for all the inline tool groups	**# show inline-tool-group traffic-rate all**

Task	Command
that are part of the inline flow deployment.	
Displays the Rx and Tx statistics for the specified inline tool group alias that is part of the inline flow deployment.	**# show inline-tool-group traffic-rate alias <alias_name>**
Deletes a specified inline tool group.	**(config) # no inline-tool-group alias inToolGroup**
Deletes the comment for this inline tool group.	**(config) # no inline-tool-group alias inToolGroup comment**
Disables an inline tool group.	**(config) # no inline-tool-group alias inToolGroup enable**
Disables release-spare-if-possible for this inline tool group.	**(config) # no inline-tool-group alias inToolGroup release-spare-if-possible**
Deletes the spare tool for this inline tool group.	**(config) # no inline-tool-group alias inToolGroup spare-inline-tool**
Deletes the inline tool list for the specified inline tool group.	**(config) # no inline-tool-group alias inToolGroup tool-list**
Deletes all inline tool groups.	**(config) # no inline-tool-group all**

interface

Required Command-Line Mode = Configure

Use the interface command to configure settings for the eth0 Mgmt port (or the eth2 Mgmt port) on the GigaVUE-OS® HC Series node's control card.

The **interface** command has the following syntax:

```
interface <interface>
  bond <bonded interface>
  comment <comment>
  dhcp [renew]
  duplex <full | auto>
  ip address <IP address> <netmask>
  ipv6
    address <<IPv6 address>/<length> | autoconfigure> [default | privacy]
    dhcp client <enable | renew>
    enable
  mtu <MTU in bytes>
  shutdown   speed <10 | 100 | 1000 | auto>
  zeroconf
```

The following table describes the arguments for the **interface** command:

Argument	Description
interface <interface>	Configures settings for the eth0 Mgmt port, as well as the loopback (lo) interface, and eth1 and eth2 interface.
bond <bonded interface>	Adds a peer interface to a specified bonded interface. For example: **(config) # interface eth0 bond bond0**
comment <comment>	Provides a comment field for the interface, displayed whenever the interface records are listed in the CLI.
dhcp [renew]	Enables or disables DHCP on the Mgmt interface. You can also use **renew** to renew the address currently leased by the Mgmt interface. When DHCP is enabled, the Mgmt port gets its IP address and subnet mask from a DHCP server—a static address/netmask defined with the **ip address** argument is not used when DHCP is enabled. For example, the following command enables DHCP on the Mgmt port (eth0): **(config) # interface eth0 dhcp** **Note**: By default, DHCP is enabled. This means that when you run the jump-start for the first time, the default reads, "[yes]." When it is run subsequently, it will indicate yes or no based on how it was set.
duplex <full | auto>	Sets the duplex mode for the Mgmt port. You can set it to **full** or **auto** (the default). If you set it to **auto**, also set **speed** to **auto**. **Note**: Starting in software version 5.5, half-duplex support has been deprecated from the management interfaces (eth0, eth1...). If half duplex was configured in a previous software version, it will remain intact following the upgrade to 5.5 or higher release. Update to full duplex, if required.
ip address <IP address> <netmask>	Specifies a static IPv4 address and netmask for the Mgmt port. Note that defining the IP address/netmask does NOT switch the Mgmt port over from DHCP if it is currently using it. You need to explicitly disable DHCP as well with **no interface eth0 dhcp** before the static address specified here is used. You can specify the netmask using either the bit-count model or a dotted-quad (for example, **/24** or **255.255.255.0**). For example, the following command configures a static address of 192.168.1.0 with a netmask of /24: **(config) # interface eth0 ip address 192.168.1.0 /24**
ipv6 address <<IPv6 address>/<length> | autoconfigure> [default | privacy] dhcp client <enable | renew> enable	Configures IPv6 properties for the Mgmt port. You can specify a static address or accept an auto configured address. Setting the address does not enable its use—you also have to explicitly enable the IPv6 address with **interface eth0 ipv6 enable**. **Note**: If IPv6 is disabled during configuration jump-start, you have to explicitly use **ipv6 enable** command to enable the IPv6 system wide before you enable the interface configuration with **interface eth0 ipv6 enable**. • **address**—Specifies the type of IPv6 address that will be assigned to the Mgmt port. You can either explicitly configure a static address with the **address** argument or use the **autoconfigure** argument to generate one automatically. • **dhcp client—**Enables or renews DHCPv6 on this interface. • **enable**—enables the use of IPv6 for the Mgmt port.
mtu <MTU in bytes>	Specifies the MTU for the Mgmt port. Specify a value in bytes (1518 is the largest size for a standard Ethernet packet). The range is 1280–9400. The default is 1500. It is

Argument	Description
	recommended that you set the MTU value to 9400 on all platforms to avoid fragmentation. Change the MTU value for the Mgmt port only after disabling IPv6 on that particular interface using the following command: **(config) # no interface <interface> ipv6 enable**
shutdown	Administratively shuts down the Mgmt port. If you do this, you will only be able to access the node over the serial console port.
speed <10 | 100 | 1000 | auto>	Sets the speed for the Mgmt port. You can set it to **10**Mb, **100**Mb, **1000**Mb, or **auto**. Setting the **speed** to **auto** also sets **duplex** to **auto**. If you set **speed** to one of the manual settings, the system will read the last autoconfigured setting for **duplex** and set it to the manual version of the autoconfigured value.
zeroconf	Enables **zeroconf** for IPv4 on the specified interface. Enabling **zeroconf** disables DHCP and static address settings.

Related Commands

The following table summarizes other commands related to the **interface** command:

Task	Command
Displays detailed information for all interfaces.	**show interfaces**
Displays brief information for the specified interface.	**show interfaces eth0 brief**
Displays configuration information for the specified interface.	**show interfaces eth0 configured**
Deletes this interface from a bonded interface.	**(config) # no interface eth0 bond bond0**
Deletes the comment from the specified interface.	**(config) # no interface eth0 comment**
Disables DHCP for the specified interface.	**(config) # no interface eth0 dhcp**
Resets the duplex setting for the specified interface to the default (auto).	**(config) # no interface eth0 duplex**
Deletes the IP address and netmask for the specified interface.	**(config) # no interface eth0 ip address**
Deletes all IPv6 addresses for the specified interface.	**(config) # no interface eth0 ipv6 address**
Deletes IPv6 address autoconfiguration for the specified interface.	**(config) # no interface eth0 ipv6 address autoconfig**
Disables learning routes from address autoconfiguration.	**(config) # no interface eth0 ipv6 address autoconfig default**
Disables privacy extensions for address autoconfiguration.	**(config) # no interface eth0 ipv6 address autoconfig privacy**
Disables DHCPv6 on the specified interface.	**(config) # no interface eth2 ipv6 dhcp client enable**
Disables IPv6 on the specified interface.	**(config) # no interface eth0 ipv6 enable**
Resets the MTU for the specified interface to the default.	**(config) # no interface eth0 mtu**

Task	Command
Enables the specified interface.	**(config) # no interface eth0 shutdown**
Resets the speed setting for the specified interface to the default.	**(config) # no interface eth0 speed**
Disables zeroconf for the specified interface.	**(config) # no interface eth0 zeroconf**

ip

Use the **ip** command to configure TCP/IP settings for the GigaVUE-OS® HC Series node's Mgmt port, including the default gateway, DNS server, and domain name. Note that most users configure these settings using the **config jump-start** script during the initial deployment of the system. Refer to the **Hardware Installation Guide** for details.

The **ip** command has the following syntax:

```
ip
  default-gateway <next hop IP address> [interface name (eth0, eth1...)]
  dhcp
    default-gateway yield-to-static
    hostname <hostname>
    primary intf <interface name>
    send-hostname
  domain-list <domain name>
  filter
    chain <chain>
      clear
      policy <policy>
      rule <append tail | insert <rule number> | set <rule number> | modify <rule number>>
target <target>
    move <old rule number> to <new rule number>
      [comment <comment> | dest-addr <network prefix> <netmask> | dest-port <port or port
range> |
      dup-delete | in-intf <interface>| not-dest-addr <network prefix> <netmask> | not-dest-
port <port
      or port range> | not-in-intf <interface> | not-out-intf <interface> | not-protocol
<protocol> |
      not-source-addr <network prefix> <netmask> | not-source-port <port or port range> |
out-intf
      <interface> | protocol <protocol> | source-addr <network prefix> <netmask> | source-
port
      <port or port range> | state <state>]
    enable
    options include-bridges
  host <hostname> <IP address>
  map-hostname
  name-server <IPv4 or IPv6 address>
  route <network prefix> <netmask | mask length> <next hop IP address or interface name>
```

The following table describes the arguments for the **ip** command:

Argument	Description
default-gateway <next hop IP address> [interface name (eth0, eth1...)]	Specifies the default gateway for the Mgmt port (eth0). The default gateway is where the Mgmt port will send IP packets for distribution to remote networks. For example: **(config) # ip default-gateway 192.168.1.1 eth0**
dhcp **default-gateway yield-to-static** **hostname <hostname>** **primary intf <interface name>** **send-hostname**	Configures global DHCP settings as follows: • **default-gateway yield-to-static—**Preserves any statically configured default gateway instead of using an address received through DHCP. • **hostname—**Specifies the hostname to be sent during DHCP client negotiation (if **send-hostname** is enabled). • **primary intf—**Sets the interface from which non-interface-specific configuration (resolver and routes) will be accepted through DHCP. Leave this set to **eth0** (the Mgmt port). • **send-hostname—**Sends a hostname during negotiation.
domain-list <domain name>	Adds a domain name to use when resolving hostnames.
filter **chain <chain>** **clear** **policy <policy>** **rule <append tail | insert | set | modify | move>** **enable** **options include-bridges**	Configures IP filtering as follows: • **chain <chain>**—Specifies the chain. The only chains allowed are FORWARD, INPUT, and OUTPUT. o **clear**—Deletes all rules from a given chain. **Note**: The clear parameter deletes all IP filter rules, which can result in loss of connectivity between nodes in a cluster. Rather than clearing all IP filters, delete only the specific filters that are no longer required. If you use clear, the following warning is displayed: **ST1 [ST1: standby] (config) # ip filter chain FORWARD clear** **WARNING !! Clearing the ip filter INPUT chain may impact mgmt and clustering ports and operations!!** **Enter 'YES' to confirm this operation:** o **policy <policy>**—Sets the policy (the default target) for a specified chain. The only targets allowed are ACCEPT and DROP. The rules on this chain will be overrides of this default. o **rule**—Appends, inserts, sets, modifies, or moves a rule. The chains and targets allowed are the same as for policy. For details on rules, refer to **rule**. • **enable**—Enables or disables IP filtering of network traffic. The default is disabled. • **options include-bridges**—Enables or disables IP packet filtering for bridges. The default is disabled. (This is not supported.) The default policies for each chain are as follows: • OUTPUT: ACCEPT • INPUT: DROP • FORWARD: DROP For configuration examples, refer to the *IP Filter Chains for Security*.

Argument	Description
rule <append tail | insert <rule number> | set <rule number> | modify <rule number>> target <target> move <old rule number> to <new rule number> [comment <comment> dest-addr <network prefix> <netmask> dest-port <port or port range> dup-delete in-intf <interface> not-dest-addr <network prefix> <netmask> not-dest-port <port or port range> not-in-intf <interface> not-out-intf <interface> not-protocol <protocol> not-source-addr <network prefix> <netmask> not-source-port <port or port range> out-intf <interface> protocol <protocol> source-addr <network prefix> <netmask> source-port <port or port range> state <state>]	Specifies the position of a rule, which is determined by the arguments that follow **rule**, as follows: • **append tail**—Adds a new rule after all existing rules. • **insert <rule number>**—Inserts a new rule before the existing rule with the specified rule number. The specified rule number must be an existing rule. The specified rule number and all rules above it will be renumbered to make room for the new rule. • **set <rule number>**—Specifies the rule number of an existing rule and overwrites it with the new rule. • **modify <rule number>**—Modifies an existing rule at a specified rule number. • **move**—Moves an existing rule to a different position in the same chain. It is inserted at the new location, removed from the old location, and the surrounding rules are renumbered. Note the following: • Rule numbers are contiguous (there are no spaces between rule numbers). • There must always be at least one rule. • You can have multiple rules with the same target. • All of the arguments after the target are optional. The targets are as follows: • ACCEPT • DROP Netmask can be specified either as a netmask or a mask length (for example: 255.255.255.0 or /24). Dup-delete specifies that after adding or modifying a rule, delete all other existing rules that are duplicates of it. (Duplicates are otherwise not detected.) The available protocols are as follows: • tcp, udp, icmp, igmp, ah, esp, all If tcp or udp are specified, you can specify source or destination ports. State classifies the packet relative to existing connections. The states are as follows: • ESTABLISHED—means it is associated with an existing connection that has seen traffic in both directions. • RELATED—means it opens a new connection, but one that is related to an established connection. • NEW—means it opens a new, unrelated connection. You can enter more than one state by separating them with a comma.
host <hostname> <IP address>	Configures a static mapping between the specified hostname and IPv4 address. The hostname must be a valid Domain Name Service (DNS) name.
map-hostname	Enables the **map-hostname** argument to ensure a static host

Argument	Description
	mapping for the current hostname.
name-server <IPv4 or IPv6 address>	Adds another DNS name server address to the GigaVUE HC Series node's list.
route <network prefix> <netmask | mask length> <next hop IP address or interface name>	Configures a static routing entry for the GigaVUE HC Series node's Mgmt port, telling the system that any traffic destined for a particular network should be sent to a particular destination. You can specify the netmask using either the bitcount format (for example, /24) or the dotted-quad format (for example, 255.255.255.0). For example: **(config) # ip route 10.16.0.0 255.255.255.0 192.168.1.1**

Related Commands

The following table summarizes other commands related to the **ip** command:

Task	Command
Displays the active default route.	**show ip default-gateway**
Displays the configured default route.	**show ip default-gateway static**
Displays DHCP configuration information.	**show ip dhcp**
Displays IP filtering state.	**show ip filter**
Displays IP filtering state (including unconfigured rules).	**show ip filter all**
Displays IP filtering configuration.	**show ip filter configured**
Displays active routes, both dynamic and static.	**show ip route**
Displays configured static routes.	**show ip route static**
Deletes the current default route.	**(config) # no ip default-gateway**
Installs default gateway from DHCP, even if there is already a statically configured one.	**(config) # no ip dhcp default-gateway yield-to-static**
Reverts to using the system hostname for DHCP client negotiation.	**(config) # no ip dhcp hostname**
Reverts to the default interface from which non-interface-specific configuration (resolver and routes) will be accepted through DHCP.	**(config) # no ip dhcp primary-intf**
Does not send a hostname during DHCP client negotiation.	**(config) # no ip dhcp send-hostname**
Deletes a domain name.	**(config) # no ip domain-list mydomain**
Resets the policy (the default target) for a specified chain to the default.	**(config) # no ip filter chain FORWARD policy**

Task	Command
If you specify a chain and rule, deletes the rule and renumbers rules to close the gap. If you specify a chain only, deletes all the rules in that chain and resets the chain's policy to the default.	**(config) # no ip filter chain INPUT rule 3**
Disables IP filtering.	**(config) # no ip filter enable**
Does not apply IP filters to bridges. (This is not supported.)	**(config) # no ip filter options include-bridges**
Deletes static hostname/IPv4 address mappings from a specified host.	**(config) # no ip host myhost 10.10.10.10**
Deletes static hostname/IPv4 address mappings from the localhost.	**(config) # no ip host localhost 10.10.10.10**
Does not ensure a static host mapping for the current hostname.	**(config) # no ip map-hostname**
Deletes a name server using IPv4 or IPv6 address.	**(config) # no ip name-server 1.1.1.1**
Deletes a static route.	**(config) # no ip route 0.0.0.0 /21**

ip interface

Required Command-Line Mode = Configure

Use the **ip interface** command to configure an IP interface with network/tool/circuit port.

IP interfaces with tool or network ports are used together with GigaSMART® operations, including the encapsulation and decapsulation components, to send and receive traffic. Refer to the "*Working with GigaSMART Operations*" section in the *GigaVUE Fabric Management Guide* for details.

The **ip interface** command has the following syntax:

```
ip interface alias <alias>
  attach <port-id>
  comment <comment for the ip interface>
  ip address <ip address> <netmask | mask length>
  ipv6 address <IPv6 address>/<len>
  gw <gw address>
  gw-ipv6 <ipv6 gw address>
  mtu <mtu value in bytes>
  gsgroup
    add <gsgroup-alias>
    delete <gsgroup-alias>
  netflow-exporter
    add <netflow-exporter-alias>
    delete <netflow-exporter-alias>
```

The following table describes the arguments for the **ip interface** command:

Argument	Description
ip interface alias <alias>	Specifies the IP interface alias on the GigaVUE HC Series node to be used as the encapsulation or decapsulation port.
attach <port-id>	Specifies the port ID that will be attached to the IP interface.
<comment for the ip interface>	Specifies the description of the IP interface.
ip address <IP address> <netmask | mask length>	Specifies the IPv4 address and subnet mask to be used for the IP interface. You can specify the subnet mask using either of the following formats: • **netmask**—For example, 255.255.255.248 • **mask length**—For example, /29
ipv6 address <IPv6 address>/<len>	Specifies the IPv6 address and prefix length to be used for the IP interface. For example, 2001::1 /64
gw <gw address>gw-ipv6 <ipv6 gw address>	Specifies the IPv4 or IPv6 address of the default gateway for this IP interface.
mtu	Specifies the MTU size for the IP interface. The MTU is fixed at 9400 for all network/tool ports on the following platforms: • GigaVUE-HC1 • GigaVUE-HC3 • GigaVUE-TA25, GigaVUE-TA100, ,and GigaVUE-TA200. **RECOMMENDATION:** Set the MTU to 9400 on all platforms to avoid fragmentation.
gsgroup **add <gsgroup-alias>** **delete <gsgroup-alias>**	Specifies the GigaSMART group alias that will be attached or deleted from this IP interface.
netflow-exporter **add <netflow-exporter-alias>** **delete <netflow-exporter-alias>**	Specifies the NetFlow exporter alias that will be associated with this IP interface or deleted from this IP interface.

Example

(config) # port 1/1/g1 type tool **(config) # port 1/1/g1 type network** **(config ip interface alias test) #** **(config ip interface alias test) # attach 1/1/g1** **(config ip interface alias test) # ip address 1.1.1.1 /29** **(config ip interface alias test) # ipv6 address ::/0** **(config ip interface alias test) # gw 1.1.1.2**	Attaches the port 1/1/g1with the IP interface alias “test” and designates the port as a network port, configures its IPv6 address, prefix length, gateway, and MTU size, and assigns a GigaSMART group and NetFlow exporter to it.

(config ip interface alias test) # mtu 9400 **(config ip interface alias test) # gsgroup add gsg** **(config ip interface alias test) # netflow-exporter add exp1,exp2** **(config ip interface alias test) # exit**	

The following table summarizes other commands related to the **ip interface** command:

Task	Command
Displays all IP interfaces.	**# show ip interfaces**
Displays a specified IP interface.	**# show ip interface alias test**
Displays destinations for a specified IP interface.	**# show ip destination interface-alias <alias>**
Displays IP interface statistics.	**# show ip interface stats**
Displays statistics for a specific IP interface.	**# show ip interface stats alias <alias>**
Displays destination statistics for a specific IP interface.	**# show ip destination stats interface-alias <alias>**
Displays IP destination status that are filtered by GigaSMART groups.	**# show ip destination gsgroup <gsgroup-alias>**
Displays IP destination statistics that are filtered by GigaSMART groups.	**# show ip destination stats gsgroup <gsgroup-alias>**
Deletes all IP interfaces.	**(config) # no ip interface all**
Deletes the GigaSMART group that is associated with the IP interface.	**(config) # ip interface alias giga_auto_tunnel_1_1_g1 gsgroup delete gsg,gsg1** **exit**

ipv6

Use the **ipv6** command to configure IPv6 settings for the GigaVUE-OS® HC Series node's Mgmt port, including enabling the use of IPv6, setting the default IPv6 gateway, and configuring static mappings and routes for IPv6. Note that most users configure these settings using the **config jump-start** script during the initial deployment of the system. Refer to the **Hardware Installation Guide** for details.

The **ipv6** command has the following syntax:

```
ipv6
  default-gateway <next hop IP address or interface name> <eth0, eth1...>
  dhcp
    primary intf <interface name>
    stateless
  enable
  filter
    chain <chain>
```

```
        clear
        policy <policy>
        rule <append tail | insert <rule number> | set <rule number> | modify <rule number>>
target <target>
          move <old rule number> to <new rule number>
            [comment <comment> | dest-addr <network prefix> <netmask> | dest-port <port or
port range> |
              dup-delete | in-intf <interface>| not-dest-addr <network prefix> <netmask> | not-
dest-port
              <port or port range> | not-in-intf <interface> | not-out-intf <interface> | not-
protocol <protocol> |
              not-source-addr <network prefix> <netmask> | not-source-port <port or port
range> |
              out-intf <interface> | protocol <protocol> | source-addr <network prefix>
<netmask> |
              source-port <port or port range> | state <state>]
      enable
      options include-bridges
    host <hostname> <IPv6 address>
    map-hostname
    neighbor <IPv6 address> <interface name> <MAC address>
    route <IPv6 prefix> <next hop IPv6 address or interface name> [eth0, eth1...]
```

The following table describes the arguments for the **ipv6** command:

Argument	Description
default-gateway <next hop IP address or interface name> <eth0, eth1...>	Sets the default IPv6 gateway for the specified interface (eth0-Mgmt, eth1-Stacking, or lo-loopback).
dhcp **primary intf <interface name>** **stateless**	Configures global DHCP settings as follows: • **primary intf**—Sets the interface from which non-interface-specific configuration (resolver and routes) will be accepted through DHCP. Leave this set to **eth0** (the Mgmt port). • **stateless—**Enables or disables stateless DHCP requests. Stateless information is mainly DNS configuration, so this option excludes getting an IPv6 address from the server.
enable	Enables the use of IPv6 generally. Both IPv6 and IPv4 can be enabled at the same time.
filter **chain <chain>** **clear** **policy <policy>** **rule <append tail | insert | set | modify | move>** **enable** **options include-bridges**	Configures IP filtering as follows: • **chain <chain>**—Specifies he chain. The only chains allowed are FORWARD, INPUT, and OUTPUT. o **clear**—Deletes all rules from a given chain. **Note**: The clear parameter deletes all IP filter rules, which can result in loss of connectivity between nodes in a cluster. Rather than clearing all IP filters, delete only the specific filters that are no longer required. If you use clear, the following warning is displayed: **ST1 [ST1: standby] (config) # ip filter chain FORWARD clear**

Argument	Description
	WARNING !! Clearing the ip filter INPUT chain may impact mgmt and clustering ports and operations!! **Enter 'YES' to confirm this operation:** o **policy <policy>**—Sets the policy (the default target) for a specified chain. The only targets allowed are ACCEPT and DROP. The rules on this chain will be overrides of this default. o **rule**—Appends, inserts, sets, modifies, or moves a rule. The chains and targets allowed are the same as for policy. For details on rules, refer to **rule**. • **enable**—Enables or disables IP filtering of network traffic. The default is disabled. • **options include-bridges**—Enables or disables IP packet filtering for bridges. The default is disabled. (This is not supported.) The default policies for each chain are as follows: • OUTPUT: ACCEPT • INPUT: DROP • FORWARD: DROP For configuration examples, refer to IP Filter Chains for Security.
rule <append tail | insert <rule number> | set <rule number> | modify <rule number>> target <target> move <old rule number> to <new rule number> **comment <comment>** **dest-addr <network prefix> <netmask>** **dest-port <port or port range>** **dup-delete** **in-intf <interface>** **not-dest-addr <network prefix> <netmask>** **not-dest-port <port or port range>** **not-in-intf <interface>** **not-out-intf <interface>** **not-protocol <protocol>** **not-source-addr <network prefix> <netmask>** **not-source-port <port or port range>** **out-intf <interface>** **protocol <protocol>** **source-addr <network prefix> <netmask>** **source-port <port or port range>** **state <states>**	Specifies the position of a rule, which is determined by the arguments that follow **rule**, as follows: • **append tail**—Adds a new rule after all existing rules. • **insert <rule number>**—Inserts a new rule before the existing rule with the specified rule number. The specified rule number must be an existing rule. The specified rule number and all rules above it will be renumbered to make room for the new rule. • **set <rule number>**—Specifies the rule number of an existing rule and overwrites it with the new rule. • **modify <rule number>**—Modifies an existing rule at a specified rule number. • **move**—Moves an existing rule to a different position in the same chain. It is inserted at the new location, removed from the old location, and the surrounding rules are renumbered. Note the following: • Rule numbers are contiguous (there are no spaces between rule numbers). • There must always be at least one rule. • You can have multiple rules with the same target. • All of the arguments after the target are optional. The targets are as follows: • ACCEPT • DROP Netmask can be specified either as a netmask or a mask length

Argument	Description
	(for example: 255.255.255.0 or /24). Dup-delete specifies that after adding or modifying a rule, delete all other existing rules that are duplicates of it. (Duplicates are otherwise not detected.) The available protocols are as follows: • tcp, udp, icmp, igmpv6, ah, esp, all If tcp or udp are specified, you can specify source or destination ports. State classifies the packet relative to existing connections. The states are as follows: • ESTABLISHED—means it is associated with an existing connection that has seen traffic in both directions. • RELATED—means it opens a new connection, but one that is related to an established connection. • NEW—means it opens a new, unrelated connection. You can enter more than one state by separating them with a comma.
host <hostname> <IPv6 address>	Configures a static mapping between the specified hostname and IPv6 address. The hostname must be a valid Domain Name Service (DNS) name. **Note**: IPv6 must be enabled before you can configure an IPv6 syslog server. If IPv6 is not enabled, the following error message is displayed: **(config) # logging fdc9:5895:4203:95c0:4203:95c0:4146:36cc** **% IPv6 is disabled. IPv6 address/dns name fdc9:5895:4203:95c0:4203:95c0:4146:36cc is not allowed.** For example: **(config) # ipv6 enable** **(config) # ipv6 host syslog.ipv6 fdc9:5895:4203:95c0:4203:95c0:4146:36cc**
map-hostname	Specifies a static IPv6 host mapping for the current hostname.
neighbor <IPv6 address> <interface name> <MAC address>	Configures static IPv6/MAC (link layer) neighbor pairs for eth0 or eth1.
route <IPv6 prefix> <next hop IPv6 address or interface name> [eth0, eth1...]	Adds an IPv6 static route by specifying the nexthop interface name or IPv6 address for a particular IPv6 network prefix. For example: **(config) # ipv6 route 2001:db8:701f::/48 fdc9:5895:4203:95c0:4203:95c0:4146:36cc eth0**

Related Commands

The following table summarizes other commands related to the **ipv6** command:

Task	Command
Displays IPv6 information.	**# show ipv6**
Displays active IPv6 default routes.	**show ipv6 default-gateway**
Displays configured IPv6 default routes.	**show ipv6 default-gateway static**
Displays DHCP configuration information.	**show ipv6 dhcp**
Displays IP filtering configuration or status.	**show ipv6 filter**
Displays IP filtering state (including unconfigured rules).	**show ipv6 filter all**
Displays IP filtering configuration.	**show ipv6 filter configured**
Displays all IPv6 neighbor information, which is part of the IP interface configuration. (The configuration is done as part of the IP interface in GigaVUE-OS with IPv6).	**show ipv6 neighbors**
Displays active IPv6 routes, both dynamic and static.	**show ipv6 route**
Displays configured static IPv6 routes.	**show ipv6 route static**
Deletes all static IPv6 default routes.	**(config) # no ipv6 default-gateway**
Reverts to the default interface from which non-interface-specific (resolver) configuration will be accepted through DHCPv6.	**(config) # no ipv6 dhcp primary-intf**
Disables stateless DHCPv6 requests (request all information, including an IPv6 address).	**(config) # no ipv6 dhcp stateless**
Disables IPv6 for the entire system.	**(config) # no ipv6 enable**
Resets the policy (the default target) for a specified chain to the default.	**(config) # no ipv6 filter chain FORWARD policy**
If you specify a chain and rule, deletes the rule and renumbers rules to close the gap. If you specify a chain only, deletes all the rules in that chain and resets the chain's policy to the default.	**(config) # no ipv6 filter chain INPUT rule 2**
Disables IP filtering for IPv6.	**(config) # no ipv6 filter enable**
Does not apply IP filters to bridges. (This is not supported.)	**(config) # no ipv6 filter options include-bridges**
Deletes static hostname/IPv6 address mappings.	**(config) # no ipv6 host localhost6**
Does not ensure a static host mapping for current hostname.	**(config) # no ipv6 map-hostname**
Deletes static IPv6 neighbor MAC (link layer) address mappings.	**(config) # no ipv6 neighbor fe80::209:fff:fe4a:e5ce eth0 00:09:0F:4A:E5:CE**
Deletes an IPv6 static route.	**(config) # no ipv6 route ::/0**

job

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **job** command to configure scheduled jobs.

The **job** command has the following syntax:

```
job <job ID>
   command <sequence #> <CLI command>
   comment <string>
   enable
    execute
    fail-continue
    name <friendly name>
   schedule type <daily | monthly | once | periodic | type | weekly>
```

The following table describes the arguments for the **job** command:

Argument	Description
<job ID>	Specifies an ID number of a job.
command <sequence #> <CLI command>	Adds a CLI command to a job. The **sequence #** is an integer that controls the order in which commands are executed relative to other commands in the job. For example: **(config) # job 12 command 1 "map alias recordtodiskmap from 1/1/x1..x10"** **NOTE:** Commands are not validated when a job is configured. To avoid a problem during execution, a good practice is to do a dry run first.
comment <string>	Adds a comment to a job. For example: **(config) # job 12 comment "weekly job"** show jobs command.
enable	Enables or disables a job. For example: **(config) # job 12 enable** If a job is disabled, it will not be executed automatically according to its schedule. Also, if a job is disabled, it cannot be executed manually.
execute	Forces the immediate execution of a job. For example: **(config) # job 12 execute** The use of this command does not affect the execution of a scheduled job. **NOTE:** To execute immediately, the job must be enabled.
fail-continue	Continues execution of a job, regardless of any failures. For example: **(config) # job 12 fail-continue**

Argument	Description
	By default, a job halts as soon as any command in the job fails.
name <friendly name>	Configures a friendly name for a job. For example: **(config) # job 12 name MyJob**
schedule type <daily | monthly | once | periodic | weekly>	Configures the type of schedule on which a job automatically executes. The schedule types are as follows: • **daily**—Executes the job daily at a specified time. For details on daily schedules, refer to **schedule daily**. • **monthly**—Executes the job monthly on a specified day of the month. For details on monthly schedules, refer to **schedule monthly**. • **once**—Executes the job once at a single specified date and time. For details on one-time schedules, refer to **schedule once**. • **periodic**—Executes the job periodically on a specified fixed time interval, beginning at a fixed point in time. For details on periodic schedules, refer to **schedule periodic**. • **weekly**—Executes the job weekly at a specified time on one or more specified weekdays. For details on weekly schedules, refer to **schedule weekly**. • The default is **once**. Each schedule type requires additional parameters. Refer to **schedule**.
schedule daily time <hh>:<mm>:<ss> | schedule daily start date <yyyy>/<mm>/<dd> schedule daily end date <yyyy>/<mm>/<dd>	Sets the time at which a daily job will execute every day, or sets a date range within which the daily job is eligible to execute. For example: **(config) # job 12 schedule daily time 10:02:22** **(config) # job 12 schedule daily start date 2014/12/25** **(config) # job 12 schedule daily end date 2014/12/26** For **start** and **end**, you can specify an absolute start and end date.
schedule monthly day-of-month<day> | schedule monthly time <hh>:<mm>:<ss> | schedule monthly interval <months> | schedule monthly start date <yyyy>/<mm>/<dd> schedule monthly end date <yyyy>/<mm>/<dd>	Sets the day of the month at which a job will execute monthly, or sets the time of day at which the job will execute on the day of the month specified, or sets the number of months between executions of the monthly job, or sets a date range within which the monthly job is eligible to execute. For example: **(config) # job 12 schedule monthly day-of-month 15** **(config) # job 12 schedule monthly time**

Argument	Description
	10:03:22 **(config) # job 12 schedule monthly interval 3** **(config) # job 12 schedule monthly start date 2014/12/2** **(config) # job 12 schedule monthly end date 2014/12/2** For **day-of-month**, you can specify days as a number from 1 to 28. (Days 29-31 are not allowed.) However, you can also specify days as a number from -1 to -28, which is the number of days before the first of the following month. For example, -1 means the last day of the month, -2 means the second to last day of the month. For **interval**, you can specify the number of months between executions. For example, 2 means every 2 months. For **start** and **end**, you can specify an absolute start and end date.
schedule once time <hh>:<mm>:<ss> [date <yyyy>/<mm>/<dd>]	Sets the date and time at which a job will execute once. For example: **(config) # job 12 schedule once time 10:03:22 date 2014/12/25**

Argument	Description
schedule periodic interval <time interval> | schedule periodic start date <yyyy>/<mm>/<dd> [time <hh>:<mm>:<ss>] schedule periodic end date <yyyy>/<mm>/<dd> [time <hh>:<mm>:<ss>]	Sets the time interval between executions of the periodic job, or sets the date and time range within which the periodic job is eligible to execute. For example: **(config) # job 12 schedule periodic interval 2h3m4s** **(config) # job 12 schedule periodic start date 2014/12/25 time 10:03:22** **(config) # job 12 schedule periodic end date 2014/12/2 time 10:03:30** For **start** and **end**, you can specify an absolute start and end date, and absolute start and end time.
schedule weekly time <hh>:<mm>:<ss> | schedule weekly time day-of-week <sun | mon | tue | wed | thu | fri | sat> | schedule weekly start date <yyyy>/<mm>/<dd> schedule weekly end date <yyyy>/<mm>/<dd>	Sets the time at which a job will execute weekly, or sets the days of the week on which the job will execute, or sets a date range within which the weekly job is eligible to execute. For example: **(config) # job 12 schedule weekly time 10:03:22** **(config) # job 12 schedule weekly day-of-week mon** **(config) # job 12 schedule weekly start date 2014/12/25** **(config) # job 12 schedule weekly end date 2014/12/2** For **time**, the single time specified under **weekly** applies to all selected days. For **day-of-week**, you can specify more than one day, but each day cannot be specified more than once. For **start** and **end**, you can specify an absolute start and end date.

Related Commands

The following table summarizes other commands related to the **job** command:

Task	Command
Displays the configuration and state of all jobs. **NOTE:** Job state information is lost on reboot.	**# show jobs**
Displays the configuration and state of a specified job, including the results of the last execution.	**# show jobs 12**
Deletes a specified job.	**(config) # no job 12**
Deletes a command from a job.	**(config) # no job 12 command 1**

Task	Command
Deletes a comment associated with a job.	**(config) # no job 12 comment**
Disables a job.	**(config) # no job 12 enable**
Returns execution to the default behavior.	**(config) # no job 12 fail-continue**
Deletes the name of a job.	**(config) # no job 12 name**
Resets the schedule type to its default, which is **once**.	**(config) # no job 12 schedule type**
Resets the schedule type **once** to its default time, which is midnight on January 1, 1970 (which means no automatic execution).	**(config) # no job 12 schedule once time**
Resets the **daily** schedule to its default time, which is midnight.	**(config) # no job 12 schedule daily time**
Resets the **weekly** schedule to its default time, which is midnight.	**(config) # no job 12 schedule weekly time**
Resets the **monthly** schedule to its default day, which is 1.	**(config) # no job 12 schedule monthly day-of-month**
Resets the monthly **interval** to its default, which is 1.	**(config) # no job 12 schedule monthly interval**
Resets the periodic **interval** to its default, which is 1 hour.	**(config) # no job 12 schedule periodic interval**
Schedules the execution of a script, for example, to apply a configuration text file.	**(config) # job 12 command 1 "configuration text file addstoragemap.txt apply fail-continue"**
Schedules the execution of a script, for example, to upload a configuration text file. **NOTE:** For uploading configuration and other files, generating a public key for authentication is recommended.	**(config) # job 12 command 1 "configuration text generate active running upload scp://qa:qa@1.1.1.1/daily"**
Schedules a job to run at night. For example, you might want to turn off monitoring at night because your network is doing lots of backups or storage maintenance. The next morning, you can run the following job.	**(config) # job 12 command 1 "no map alias recordtodiskmap"** **(config) # job 13 command 1 "map alias recordtodiskmap from 1/1/x1..x10"**

ldap

Required Command-Line Mode = ConfigureRequired User Level = Admin

Use the **ldap** command to specify the LDAP servers to be used for authentication. You can specify multiple LDAP servers. Servers are used as fallbacks in the same order they are specified—if the first server is unreachable, the second server is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

Refer to the *"LDAP"* section in the *GigaVUE Fabric Management Guide* for examples of adding and configuring an LDAP server.

The **ldap** command has the following syntax:

```
ldap
   base-dn <string>
  bind-dn <string>
  bind-password <string> | 9 <encrypted string>
  extra-user-params roles enable
  group-attribute <<string> | member | uniqueMember>
  group-dn <string>
  host <IPv4/IPv6 address or hostname> [order <order number> | last]
  login-attribute <<string> | uid | sAMAccountName>
  port <port number>
  referrals
  remote-user-group
    base-dn <base-dn string> map-to <local account>
    map <disable | enable>
  scope <one-level | subtree>
  ssl ca-list <none | default-ca-list>
    cert-verify mode <none | ssl | tls>
    ssl-port <port number>
  timeout-bind <seconds>
  timeout-search <seconds>
  version <2 | 3>
```

The following table describes the arguments for the **ldap** command. The **key**, **retransmit**, and **timeout** values can be specified both globally and on a per-host basis. Per-host values override any configured global values.

Argument	Description
base-dn <string>	Identifies the base distinguished name (location) of the user information in the LDAP server's schema. Specify this by identifying the organizational unit (ou) in the base DN. Provide the value as a string with no spaces. For example: **(config) # ldap base-dn "ou=People,dc=mycompany,dc=com"** This is a global setting. It cannot be configured on a per-host basis.
bind-dn <string>	Specifies the distinguished name (dn) on the LDAP server with which to bind. By default, this is left empty for anonymous login. This is a global setting. It cannot be configured on a per-host basis.
bind-password <string> | 9 <encrypted string>	Provides the credentials to be used for binding with the LDAP server. If **bind-dn** is undefined for anonymous login (the default), **bind-password** should also be undefined. This is a global setting. It cannot be configured on a per-host basis. To configure an encrypted bind-password, use 9 keyword. **(config) # ldap bind-password 9 8IEiQIlyQ1JhfkfhFBgAds/yjxHlhvTCkEakIWJivnA=**
extra-user-params roles	Enables the GigaVUE HC Series node to accept user roles assigned in the LDAP

Argument	Description
enable	server. Refer to the "Granting Roles with External Authentication Servers" in the GigaVUE Fabric Management Guide for details.
group-attribute <<string> | member | uniqueMember>	Specifies the name of the attribute to check for group membership. If you specify a value for **group-dn**, the attribute you name here will be checked to see whether it contains the user's distinguished name as one of the values in the LDAP server. This is a global setting. It cannot be configured on a per-host basis.
group-dn <string>	Specifies that membership in the named **group-dn** is required for successful login to theGigaVUE HC Series node. By default, the **group-dn** is left empty—group membership is not required for login to the system. If you do specify a **group-dn**, the attribute specified by the **group-attribute** argument must contain the user's distinguished name as one of the values in the LDAP server or the user will not be logged in. This is a global setting. It cannot be configured on a per-host basis.
host <IPv4/IPv6 address or hostname> [order <order number> | last]	Specifies the IP address (IPv4 or IPv6) or hostname of the LDAP server where authentication requests will be sent. Examples: **(config) # ldap host 192.168.1.225** **(config) # ldap host 2001:db8:a0b:12f0::66** **(config) # ldap host www.MyCo.com** Servers are tried in the same order they are added to the list. Check the current order with the **show ldap** command. Then, use the **host** command with the **order** argument to change the order, if necessary. You can either specify a new order number for a host or move it to the bottom of the list with **order last**. For example: **(config) # ldap host 192.168.1.225 order last**
login-attribute <<string> | uid | sAMAccountName>	Specifies the name of the LDAP attribute containing the login name. The default is sAMAccountName. You can also specify a custom string or **uid** (for User ID). This is a global setting. It cannot be configured on a per-host basis.
port <port number>	Specifies the port number on which the LDAP server is running. If you do not specify a port, the default LDAP authentication port number of 389 is used. This is a global setting. It cannot be configured on a per-host basis.
referrals	Enables LDAP referrals. If an LDAP server does not have a requested object, it can return a referral to another destination. You can toggle this option using **no ldap referrals** to specify whether the GigaVUE HC Series node should accept the referral and query the suggested server.
remote-user-group base-dn <base-dn string> map-to <local account> map <disable | enable>	Maps a remote user group to a local user account as follows: • **base-dn**—Specifies the base-dn of the remote user group. First specify the base-dn string, then the **map-to** keyword followed by the local account name. • **map**—Enables or disables the mapping policy of the remote user group. Examples: **(config) # ldap remote-user-group map enable** **(config) # ldap remote-user-group base-dn**

Argument	Description
	"CN=gvhd,OU=gigamontaps,DC=gigamondev,DC=com" map-to admin **(config) # ldap remote-user-group base-dn "CN=gvhd1,OU=gigamontaps,DC=gigamondev,DC=com" map-to admin** **Note**: If a user account exists on the remote server as well as on the local device, the remote user will be mapped to the local account, regardless of the LDAP mapping policy.
scope <one-level | subtree>	Specifies the search scope for the user under the base distinguished name (dn): • **subtree**—Searches the base dn and all of its children. This is the default. • **one-level**—Searches only the immediate children of the base dn. This is a global setting. It cannot be configured on a per-host basis.
ssl ca-list <none | default-ca-list> cert-verify mode <none | ssl | tls> ssl-port <port number>	Configures the GigaVUE HC Series node's use of SSL for communications with LDAP servers as follows: • **ca-list**—Configures LDAP to use a supplemental CA list. Set to **default-ca-list** to use the CA list configured with the **crypto** command. Set to **none** if you do not want to use a supplemental list. • **cert-verify**—Enables LDAP SSL/TLS certificate verification. Use **no ssl cert-verify** to disable. • **mode**—Enables SSL or TLS to secure communications with LDAP servers as follows: o **none**—Does not use SSL or TLS to secure LDAP. o **ssl**—Secures LDAP using SSL over the SSL port. o **tls**—Secures LDAP using TLS over the default server port. • **ssl-port**—Configures LDAP SSL port number
timeout-bind <seconds>	Specifies how long the GigaVUE HC Series node should wait for a response from an LDAP server to a bind request before declaring a timeout failure. The valid range is 0-60 seconds. The default is 5 seconds.
timeout-search <seconds>	Specifies how long the GigaVUE HC Series node should wait for a response from the LDAP server to a search request before declaring a timeout failure. The valid range is 0-60 seconds. The default is 5 seconds.
version <2 | 3>	Specifies the version of LDAP to use. The default is version 3, which is the current standard. Some older servers still use version 2. This is a global setting. It cannot be configured on a per-host basis.

Related Commands

The following table summarizes other commands related to the **ldap** command:

Task	Command
Displays the list of configured LDAP servers and related LDAP settings.	**# show ldap**
Resets user search base.	**(config) # no ldap base-dn**

Task	Command
Deletes DN to which to bind to the server.	**(config) # no ldap bind-dn**
Deletes bind credentials.	**(config) # no ldap bind-password**
Does not allow the LDAP server to include additional roles for a remotely authenticated user in the response.	**(config) # no ldap extra-user-params roles enable**
Resets group membership attribute to use default (member).	**(config) # no ldap group-attribute**
Deletes the distinguished name group required for authorization. The default is no authorization checks.	**(config) # no ldap group-dn**
Stops sending LDAP authentication requests to host with specified IPv4 or IPv6 address, or hostname.	**(config) # no ldap host 1.1.1.1** **(config) # no ldap host www.MyCo.com**
Resets login name attribute to use the default.	**(config) # no ldap login-attribute**
Resets LDAP server port number to the default (389).	**(config) # no ldap port**
Disables LDAP referrals.	**(config) # no ldap referrals**
Deletes the mapping of a remote user group to a local account.	**(config) # no ldap remote-user-group base-dn "ou=People,dc=mycompany,dc=com" map-to monitor**
Resets user search scope to the default (subtree).	**(config) # no ldap scope**
Disables the use of a supplemental CA certificates list.	**(config) # no ldap ssl ca-list**
Disables LDAP SSL/TLS certificate verification.	**(config) # no ldap ssl cert-verify**
Resets LDAP SSL/TLS mode to the default.	**(config) # no ldap ssl mode**
Resets LDAP SSL port number to the default.	**(config) # no ldap ssl ssl-port**
Resets LDAP timeout for binding to a server.	**(config) # no ldap timeout-bind**
Resets LDAP timeout for searching for user information.	**(config) # no ldap timeout-search**
Resets LDAP version to the default.	**(config) # no ldap version**

license

Required Command-Line Mode = Configure

Use the **license** command to enable additional GigaVUE-OS features. Some license keys enable GigaSMART applications. Other license keys enable ports or clustering.

The following products have GigaSMART licensing:

- GigaVUE-HC1
- GigaVUE-HC1-Plus

- GigaVUE-HC3
- GigaVUE-HCT

For GigaSMART licensing details, refer to the "*Working with GigaSMART Operations*" chapter in the *GigaVUE Fabric Management Guide*.

The following products have port licensing:

- GigaVUE-TA25A (a 24-port version of GigaVUE-TA25)
- GigaVUE-TA100
- GigaVUE-TA200
- GigaVUE-TA200E
- GigaVUE-TA400E
- Certified Traffic Aggregation White Box

NOTE: GigaVUE-TA25 have all ports enabled.

The following products have Advanced Features License:

- GigaVUE-TA25 and GigaVUE-TA25A
- GigaVUE-TA25E
- GigaVUE-TA100
- GigaVUE-TA200
- GigaVUE-TA200E
- GigaVUE-TA400
- GigaVUE-TA400E

For port licensing and Advanced Features License details, refer to the respective **Hardware Installation Guide**.

The **license** command has the following syntax:

(config) # license install box-id <box ID> key <license key>

For example:

(config) # license install box-id 1 key LK2-SMT_HC0_R-7YF0-QL2M-1G5L-Q32C-T27X-C0VU-CD5H-NJUK-77XC-0UB1-EDMN-JUK7-7XC0-W3JC-5LNQ-RBJ7-XHY1-T7AU-KECM-N6JU-K741-6L2G-RW60-Q3LC-A479-0L6E-HH70-W30E-9T8G-V20Q-UFEM-P78F-9Q86-GT6B-BH3Y-N8QQ-9H20-056C-BHQQ-8KUV

The key is generated by Gigamon. It consists of a long string beginning with LK2, which is a protocol, followed by the card or module (SMT_HC0_R), followed by the content of the license key.

The following table describes the arguments for the **license** command:

Argument	Description
box-id <box ID>	Configures the system's box ID. The box ID identifies the node in the system.
key <license key>	Configures the license key.
revoke key <license key>	Revokes the license key. Once the license is revoked, the same key cannot be used for the same device at anytime.

Related Commands

The following table summarizes other commands related to the **license** command:

Task	Command
Displays all installed licenses.	**# show license** **Note**: The license type is displayed in the **show license** command output. For licenses obtained before 5.7 release, the license type field is displayed as NA.
Displays the installed licenses on a specified node.	**# show license box-id 1**

map

Required Command-Line Mode = Configure

Use the **map** command to configure flow maps and map parameters.

NOTE: SNAP/LLC packets are not supported in GigaVUE-TA400.

The **map** command has the following syntax:

```
map alias <alias>
  a-to-b <<ordered list of inline tools and inline tool groups> | bypass | same | reverse>
  b-to-a <<ordered list of inline tools and inline tool groups> | bypass | same | reverse>
  comment <comment>
  enable
 encap-tunnel <tunnel name>
  flowrule
    add <drop | pass> gtp <imsi | imei | msisdn> <number[*]> [comment <comment> | interface
<Gn | S11 |
      S5 | S10> | version <1 | 2>]
    delete <all | rule-id <rule ID>>
  flowsample
fstype
 default
 rotational timer <value> offset <value>
 flowsample
```

add 5g<dnn<pattern>> [comment <comment>] <pei<number[*]>> <supi<number[*]>>
<gpsi<number[*]>> <nsiid <SST | SST.SD>> <nci><pattern><tac><pattern>< plmn-id>
<mcc.mnc> <5qi><percentage <percentage range>>
<qci <value>> <version <1 | 2>>

add gtp <apn <pattern>> [comment <comment>] <imei <number[*]>> <imsi <number[*]>>
<interface <Gn | S11 | S5 | S10>> <msisdn <number[*]>> <eci><pattern>
<plmn-id> <mcc.mnc> <tac><pattern> <percentage <percentage range>>
<qci <value>> <tac-5g> <nci> <snssai> <version <1 | 2>>
<recalc>

add sip <caller-id <caller ID>> <percentage <percentage range>>
delete <gtp | sip> <all | priority-id <rule ID>>
insert <after | before> <priority index> <gtp> <apn <pattern>> [comment <comment>]
<imei <number[*]>> <imsi <number[*]>> <msisdn <number[*]>> <interface <Gn | S11 | S5 |
S10>>
<percentage <percentage range>> | <qci <value>> <version <1 | 2>>
insert <after | before> <priority index> <sip> <caller-id <caller ID>> <percentage
<percentage range>>
from <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-
network-alias |
inline-network-group-alias | vport-alias>
gsrule
add <drop | pass> <criteria>
delete <all | rule-id <rule ID>>
no-rule-match pass
oob-copy from <inline-network alias | through-list item> [dir <a-to-b | b-to-a>] to <tool port
list> tag <none |
as-inline>
param traffic control
priority <after <map name> | before <map name> | highest | lowest>
roles <assign | replace> <role> [to <role list>]
rule
add <drop | pass> <criteria>
copy-from template <template alias>
delete <all | rule-id <rule ID>>
edit rule-id <rule ID> <comment <comment> | drop <criteria> | pass <criteria>>
tag <<1-4000> | auto>
rewrite-dstip <x.x.x.x>
rewrite-dstmac <xxxx.xxxx.xxxx | xx:xx:xx:xx:xx:xx>
rewrite-srcip <x.x.x.x>
rewrite-srcmac <xxxx.xxxx.xxxx | xx:xx:xx:xx:xx:xx>
vlan-op add <vlan id> tpid <value>
vlan-op modify <vlan id> tpid <value>
vlan-op strip

```
to <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias
|one-arm|
inline-tool-group-alias | inline-serial-alias | bypass | vport-alias | null-port>
  encap-tunnel <tunnel-alias>
   type <firstLevel | flexInline | inline | regular | transitLevel| secondLevel>
     firstLevel [byRule]
    flexInline [byRule | collector]
    inline [byRule]
    regular [byRule]
    secondLevel [byRule | flowFilter | flowSample | flowSample-ol | flowSample-sip |
flowWhitelist |
|  flowWhitelist-ol | flowWhitelist-sip | flowWL-5g | flowSample-5G]
   use gsop <gsop alias>
  whitelist
 add 5g
<dnn <pattern>| type <supi | ran | all >
    add gtp <apn <pattern> | interface <Gn | S10 | S11 | S5> | version <1 | 2>> | type <imsi | ran |
all >
    delete all
  add sip <all | callee-id | caller-id | dest-ip | ip-addr | src-ip>
map priority <map names>
```

The following table describes the arguments for the **map** command:

Argument	Description
alias <alias>	Specifies the name of the map. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # map alias map1** **(config map alias map1) #** The following are reserved keywords that cannot be used in map aliases with any character case: • **rule** • **map** For example, using these keywords in a map alias displays an error message: **% Invalid alias 'Rule'. 'Rule' is a reserved word.** or **% Invalid alias 'MAP'. 'MAP' is a reserved word.**
a-to-b <<ordered list of inline tools and inline tool groups> | bypass | same | reverse>	For flexible inline arrangements, specifies the sequence of inline tools or inline tool groups through which the traffic will be guided between the respective inline network ports, as follows: • **ordered list of inline tools and inline tool groups**—

Argument	Description
	Specifies the list of aliases of inline tools and inline tool groups participating in the flexible inline map in the a-to-b direction, in order. The maximum number of inline tools or inline tool groups in the list is 16 for one direction. • **bypass**—Specifies the traffic be sent to bypass. • **same**—If a map has both a-to-b and b-to-a parameters, this option specifies the same value as the other parameter. • **reverse**—If a map has both a-to-b and b-to-a parameters, this option specifies the ordered list of inline tools and inline tool groups in the other parameter, but in the reverse order. Separate each alias with a comma. For example: **(config map alias flexmap1) # a-to-b IT1,IT2,IT3,IT4**
b-to-a <<ordered list of inline tools and inline tool groups> | bypass | same | reverse>	For flexible inline arrangements, specifies the sequence of inline tools or inline tool groups through which the traffic will be guided between the respective inline network ports, as follows: • **ordered list of inline tools and inline tool groups**—Specifies the list of aliases of inline tools and inline tool groups participating in the flexible inline map in the b-to-a direction, in order. The maximum number of inline tools or inline tool groups in the list is 16 for one direction. • **bypass**—Specifies the traffic be sent to bypass. • **same**—If a map has both a-to-b and b-to-a parameters, this option specifies the same value as the other parameter. • **reverse**—If a map has both a-to-b and b-to-a parameters, this option specifies the ordered list of inline tools and inline tool groups in the other parameter, but in the reverse order. Separate each alias with a comma. For example: **config# map alias flexmap2 # b-to-a IT3,IT4,IT1,IT2**
comment <comment>	Supplies an optional comment for this map. The comment will appear in **show map** output. For example: **(config map alias map1) # comment "to SanFran"**
enable	Enables the specified map. Maps with rules can be enabled or disabled. For all map enable and disable actions, only regular maps and first level maps are supported. Other maps (such as map-

Argument	Description
	passall, second level maps, and inline maps) are not supported. When a map is disabled, traffic is not passed to the tool ports. When a map is disabled, the map rules are still present, but the map is marked as disabled. Map rules consume resources even if the map the disabled. Map statistics are not updated when a map is disabled. For example: **(config) # map alias map1 enable**
encap-tunnel	Enables the encap tunnel configuration in a specified map. For example: **(config) # map alias encap-tunnel encap1**
flowrule add <drop | pass> gtp <imsi | imei | msisdn> <number[*]> [comment <comment> | interface <Gn | S11 | S5 | S10> | version <1 | 2>] delete <all | rule-id <rule ID>>	Configures map rules for GTP correlation. The arguments are as follows: • **add**—Adds a new drop or pass flowrule to match specified IMSI, IMEI, MSISDN subscriber IDs, Evolved Packet Core (EPC) interface or GTP version can also be specified. • **delete**—Deletes all flowrules or a specified flowrule in a map by its rule ID. To specify version, use the following: • 1 for v1 • 2 for v2 • To specify **any** version, do not add either version 1 or version 2 to the flowrule. To specify EPC interfaces, use the following: • Gn for Gn/Gp • S11 for S11/S1-U • S5 for S5/S8 • S10 **NOTE:** Version and interface cannot be specified in the same flowrule. Examples: **(config) # map alias map1 flowrule add pass gtp imsi 21345*** **(config) # map alias map1 flowrule add pass gtp imsi 21345* interface S5** **(config) # map alias map1 flowrule add drop gtp imsi 21345* version 1** The maximum number of GTP flowrules is 32 per map (16 pass and 16 drop rules).

Argument	Description
	The procedure for creating a GTP flowrule for specified IMSIs is as follows: Create a GigaSMART group and associate it with one or more GigaSMART engine ports. For example: **(config) # gsgroup alias gsg1 port-list 1/1/e1** Create a GigaSMART operation using **flow-ops flow-filtering gtp** and assign it to the GigaSMART group. For example: **(config) # gsop alias gtp_sf flow-ops flow-filtering gtp port-list gsg1** Create a GigaSMART virtual port and assign it to the same GigaSMART group. For example: **(config) # vport alias vp1 gsgroup gsg1** Create a first level map directing GTP traffic from physical network ports to the virtual port created in the previous step. For example: **(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # rule add pass portsrc 2123** **(config map alias to_vp) # rule add pass portsrc 2152** **(config map alias to_vp) # exit** Create a second level map that takes traffic from the GigaSMART virtual port, applies the flow-ops GigaSMART operation, matches IMSIs specified by a flowrule, and sends matching traffic to physical tool ports. For example: **map alias IMSI-list1** **(config map alias IMSI-list1) # type secondLevel flowFilter** **(config map alias IMSI-list1) # use gsop gtp_sf** **(config map alias IMSI-list1) # to 1/1/x4** **(config map alias IMSI-list1) # from vp1** **(config map alias IMSI-list1) # flowrule add pass gtp imsi 22222222222223*** **(config map alias IMSI-list1) # exit** Refer to the *"GigaSMART GTP Correlation"* section in the *GigaVUE Fabric Management Guide* for more information.
fstype	Configures map rules for GTP rotational flow sampling.

Argument	Description
default **rotational timer <value> offset <value>**	The arguments are as follows: • **default**— This option would set the flowsample to default sampling. • **rotational** — This option would enable rotational sampling by configuring the below parameters: o **timer**— Specifies the duration of the sampling interval in minutes. A subset of flows (based on the offset and sampling percentage) will be sampled during this interval. The time interval is <15 - 45000> minutes. o **offset**—Specifies the percentage point in each timer cycle at which sampling begins. The offset percentage is 1-99.
flowsample **add gtp <apn <pattern>> [comment** **<comment>] <eci><imei <number[*]>> <imsi** **<number[*]>> <interface <Gn | S11 | S5 | S10>> <msisdn <number[*]>><plmn-id> <qci <value>> <tac> <5ql <pattern>>< version <1 | 2>> <tac-5g> <nci> <snssai> <percentage <percentage range>>** **<recalc>** **delete <gtp> <all | priority-id <rule ID>>** **insert <after | before> <priority index> <gtp> <apn <pattern>> [comment <comment>] <imei <number[*]>> <imsi <number[*]>>** **<msisdn <number[*]>> <interface <Gn |** **S11 | S5 | S10>> <percentage <percentage** **range>> | <qci <pattern> <value>> <version <1 | 2>>** **<recalc>**	Configures map rules for GTP flow sampling. The arguments are as follows: • **add**—Adds a new pass flow sampling rule to a flow sampling map to match specified IMSI, IMEI, or IMSISDN subscriber IDs. Wildcard suffixes are supported on subscriber IDs. The percentage of the flow to be sampled must also be specified. In addition, Evolved Packet Core (EPC) interface or GTP version, Access Point Name (APN), or QoS Class Identifier (QCI), can also be specified to send matching traffic to desired tool ports, based on the sampling. o The *recalc* keyword enables the periodic recalculation feature, which allows the samples to be re-evaluated through all ongoing sessions at each sampling interval o To specify version, use 1 for v1 and 2 for v2. To specify **any** version, do not add either version 1 or version 2 to the flowsample rule. o To specify EPC interface types, use: Gn for Gn/Gp, S11 for S11/S1-U, S5 for S5/S8, S10. **NOTE:** Version and interface cannot be specified in the same flowsample rule. o To specify an APN, use a pattern, with or without a wildcard prefix or suffix, up to a maximum of 100 case-insensitive characters, as well as period (.), hyphen (-), and wildcard (*). APN is not supported on GigaVUE-HB1. o To specify a QCI, use a value from 0 to 255. A wildcard prefix or suffix is not supported. QCI can only be used in flow sampling map rules in combination with APN. o To specify ECI, use hexadecimal format and

Argument	Description
	supports wildcard. The maximum characters allowed are 9, and the minimum length is 4. o To specify TAC, use hexadecimal format and supports wildcard. The maximum characters allowed are 4, and the minimum characters are 2. o To specify PLMNID, MCC and MNC values are mandatory. MCC must have 3 characters. MNC values ranges between 1 to 3. MNC supports wildcard. o To specify 5QI, use a value from 1 to 255. o To specify a percentage, use the following: ▪ 1 to 100 to specify the percentage of subscribers to sample ▪ 0 to drop sampled data that matches a rule o To specify tac-5g, use a 6 byte hex value. o To specify nci, use [eNodeB-D + CellID, exlcuding MCC & MNC] o To specify snssai [nsiid is SST | SST.SD] {SD is optional param} Examples: **(config) # map alias map1 flowsample add gtp imsi 21345* percentage 30** **(config) # map alias map1 flowsample add gtp imsi 21345* interface Gn percentage 30** **(config) # map alias map1 flowsample add gtp imsi 21345* imei 66* version 2 percentage 30** **(config) # map alias map1 flowsample add gtp apn *ims* percentage 50** **(config) # map alias map1 flowsample add gtp apn *ims* qci 5 percentage 50** You can put IMEI, IMSI, and MSISDN numbers in a single rule. The rule will only be matched if the IMEI, IMSI, and MSISDN match. **NOTE:** When a **flowsample** rule includes a 15-digit IMSI, configure the percentage as 100%. Configuring any other percentage allows the device to reject packets for that IMSI. For Example: 0% to 99% - Reject Packets and 100% - Pass Packets. • **elete**—Deletes all existing rules from a flow sampling map, or specifies a rule to delete from a flow sampling map using a priority ID. For example: **(config) # map alias map1 flowsample delete gtp all** **(config) # map alias map1 flowsample delete**

Argument	Description
	gtp priority-id 2 • **insert**—Inserts a new rule into a flow sampling map either before or after a specified priority ID. A priority ID indicates the order of rules in the map. Use **before** and **after** to order the rules. The first rule has the highest priority. The syntax for an inserted flow sampling rule is the same as for **add**. Examples: **(config) # map alias map1 flowsample insert after 12 gtp imsi 22345* percentage 70** **(config) # map alias map1 flowsample insert after 12 gtp imsi 22345* interface S10 percentage 70** **(config) # map alias map1 flowsample insert before 11 gtp imsi 22345* version 1 percentage 70** **NOTE:** When a **flowsample** rule is inserted, it will appear as an addition in the output of the running configuration. **NOTE:** The maximum number of GTP flowsample rules is 20 per map. Up to ten (10) flow sampling maps can be configured per vport. Refer to the "*GigaSMART GTP Whitelisting and GTP Flow Sampling*" section in the *GigaVUE Fabric Management Guide* for more information. Configures map rules for 5G flow sampling. The arguments are as follows:
flowsample **add 5g<dnn<pattern>> [comment <comment>] <pei<number[*]>> <supi<number[*]>> <gpsi<number[*]>> <snssai <SST | SST.SD>> <nci><pattern><tac><pattern>< plmn-id> <mcc.mnc><percentage <percentage range>>** **delete 5g<dnn<pattern>> [comment <comment>] <pei<number[*]>> <supi<number[*]>> <gpsi<number[*]>> <percentage <percentage range>>** **>** **insert <after | before> <priority index> <5g> <dnn<pattern>> [comment <comment>] <pei<number[*]>> <supi<number[*]>>**	Configures map rules for 5Gb flow sampling. The arguments are as follows: • **add**—Adds a new pass flow sampling rule to a flow sampling map to match specified GPSI, PEI, SUPI, NSI, TAC, PLMN, ECI or NCI subscriber IDs. Wildcard suffixes are supported on subscriber IDs. The percentage of the flow to be sampled must also be specified. o To specify version, use 1 for v1 and 2 for v2. To specify **any** version, do not add either version 1 or version 2 to the flowsample rule. **NOTE:** Version and interface cannot be specified in the same flowsample rule. o To specify nsiid, SST value is mandatory . SD value is optional. SD value supports wildcard. The maximum value allowed for SST is 3 and SSD is 6.

Argument	Description
	○ To specify NCI, use hexadecimal format and supports wildcard. The maximum characters allowed are 9, and the minimum length is 4. ○ To specify TAC, use hexadecimal format and supports wildcard. The maximum characters allowed are 6, and the minimum characters are 2. ○ To specify PLMNID, MCC and MNC values are mandatory. MCC must have 3 characters. MNC values ranges between 1 to 3. MNC supports wildcard. ○ To specify a percentage, use the following: ▪ 1 to 100 to specify the percentage of subscribers to sample ▪ 0 to drop sampled data that matches a rule Examples: **(config) # map alias map1 flowsample add 5G nci f12345678 percentage 100** • **delete**—Deletes all the existing rules from a flow sampling map, or specifies a rule to delete from a flow sampling map using a priority ID. For example: **(config) # map alias map1 flowsample delete 5G all** • **insert**—Inserts a new rule into a flow sampling map either before or after a specified priority ID. A priority ID indicates the order of rules in the map. Use **before** and **after** to order the rules. The first rule has the highest priority. The syntax for an inserted flow sampling rule is the same as for **add**. Examples: **(config) # map alias map1 flowsample insert after 5G imsi nci f12345678 percentage 100** **NOTE:** When a **flowsample** rule is inserted, it will appear as an addition in the output of the running configuration. Refer to the "*GigaSMART 5G Whitelisting and 5G Flow Sampling*" section in the *GigaVUE Fabric Management Guide*for more information.
flowsample add sip <caller-id <caller ID>> <percentage <percentage range>> delete <sip> <all | priority-id <rule ID>> insert <after | before> <priority index> <sip>	Configures map rules for SIP flow sampling. The arguments are as follows: • **add**—Adds a new pass flow sampling rule to a flow sampling map to match specified caller IDs. Wildcard

Argument	Description
<caller-id <caller ID>> <percentage <percentage range>>	suffixes are supported. The percentage of the flow to be sampled must also be specified. For example: **(config) # map alias map1 flowsample add sip caller-id * percentage 50** • **delete**—Deletes all existing rules from a flow sampling map, or specifies a rule to delete from a flow sampling map using a priority ID. For example: **(config) # map alias map1 flowsample delete sip all** **(config) # map alias map1 flowsample delete sip priority-id 2** • **insert**—Inserts a new rule into a flow sampling map either before or after a specified priority ID. A priority ID indicates the order of rules in the map. Use **before** and **after** to order the rules. The first rule has the highest priority. The syntax for an inserted flow sampling rule is the same as for **add**. For example: **(config) # map alias map1 flowsample insert after 12 sip caller-id * percentage 50** **NOTE:** The maximum number of SIP flowsample rules is 20 per map. Refer to *"GigaSMART SIP/RTP Correlation"* in the *GigaVUE Fabric Management Guide* for details and examples.
from <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-network-alias | inline-network-group-alias | vport-alias>	Specifies the source(s) for packets matching this map. Use one of the following: • **port-id, port-alias, port-list**—Sends matching traffic from one or more network ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **gigastream-alias, gigastream-alias-list**—Sends matching traffic from the specified GigaStream. Refer to the "*GigaStream"* section in the *GigaVUE Fabric Management Guide* for details on GigaStream. • **inline-network-alias**—Sends matching traffic from the specified inline network alias. • **inline-network-group-alias**—Sends matching traffic from the specified inline network group alias. • **vport-alias**—Sends matching traffic from the virtual port associated with the GigaSMART group. **NOTE:** You can add a maximum of 324 ports, if the ports are not attached to a GigaStream. Refer to the "*Associating Inline Networks with Inline*

Argument	Description
	Tools Using Inline Maps" section in the *GigaVUE Fabric Management Guide* for details on **inline-network-alias** and **inline-network-group-alias**. For example: **(config) # map alias map1 from port1**
gsrule **add <drop | pass> <criteria> delete <all | rule-id <rule ID>>**	Adds or deletes a gsrule (GigaSMART rule). GigaSMART rules use Adaptive Packet Filtering to match specified packets in a second level map receiving traffic from a GigaSMART virtual port (vport). The overall procedure for creating a gsrule is as follows: **1.** Create a GigaSMART group and associate it with one or more GigaSMART engine ports. For example: **(config) # gsgroup alias gsg1 port-list 1/1/e1** **2.** Create a GigaSMART operation with an Adaptive Packet Filtering (**apf**) component and assign it to the GigaSMART group. For example: **(config) # gsop alias gsfil apf set port-list gsg1** **3.** Create a GigaSMART virtual port and assign it to the same GigaSMART group. For example: **(config) # vport alias vp1 gsgroup gsg1** **4.** Create a first level map directing selected traffic from physical network ports to the virtual port you created in the previous step. For example, the following map forwards all Fiber Channel over Ethernet (ethertype 8906) traffic from 1/1/x3 to the virtual port: **(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # rule add pass ethertype 8906** **(config map alias to_vp) # exit** **5.** Create a second level map that takes traffic from the GigaSMART virtual port, applies the Adaptive Packet Filtering GigaSMART operation, includes the GigaSMART rule with the filter, and sends matching traffic to physical tool ports. For example, the following second level map includes a regular expression string match at a specified offset (in this case, the offset for the destination address in a Fiber Channel over Ethernet packet). **map alias m1**

Argument	Description
	(config map alias m1) # type secondLevel byRule **(config map alias m1) # use gsop gsfil** **(config map alias m1) # to 1/1/x1** **(config map alias m1) # from vp1** **(config map alias m1) # gsrule add pass pmatch string "\xff\xff\xfe" 29** **(config map alias m1) # exit** **NOTE:** The maximum number of gsrules that can be specified in a map is 5. Refer to the "*GigaSMART Adaptive Packet Filtering (APF)*" section for more information.
no-rule-match pass	Specifies what to do with traffic that does not match any rule in a map that only has drop rules. This argument changes the default behavior of drop to pass in a drop-only map. If you do not use this argument and there are only drop rules in a map, the default behavior is that all traffic not matching the rules will be dropped, or, if a shared collector is configured, traffic will be sent to the shared collector. However, if you use this argument and there are only drop rules in a map, traffic will be passed rather than dropped. For example: **(config) # map alias m1** **(config map alias m1) # type regular byRule** **(config map alias m1) # from 1/1/x1** **(config map alias m1) # to 2/1/x2** **(config map alias m1) # rule add drop ipver 4** **(config map alias m1) # no-rule-match pass** **(config map alias m1) # exit** When managing map rule resources, note that using this argument consumes one extra map rule.
oob-copy from <inline-network alias | through-list item> [dir <a-to-b | b-to-a>] to <tool port list> tag <none | as-inline>	For flexible inline arrangements, configures an out-of-band (OOB) map by copying from a flexible inline map as follows: • **from**—Specifies the OOB copy source as follows: o inline network alias—Taps traffic from the source inline network of the flexible inline map. o through list item—Taps traffic from a tool member in the **a-to-b** or **b-to-a** list. **NOTE:** All sources of an OOB copy configuration

Argument	Description
	must be a member of the flexible inline map, either an inline network in the **from** parameter or a single member of the **a-to-b** or **b-to-a** inline tool list. • **dir**—Specifies the direction of the source from which to tap traffic as follows: o **a-to-b**—Taps traffic from the **a-to-b** side of the source. o **b-to-a**—Taps traffic from the **b-to-a** side of the source. • **to**—Specifies the destination inline tools. The **to** parameter can be a regular tool port, a hybrid port, or a GigaStream on the same GigaVUE-OS node. • **tag**—Specifies the OOB copy tag as follows: o **none**—Does not tag packets going to the OOB tool. The default is **none**. o **as-inline**—Uses the same external VLAN tag as the flexible inline map. For example: **(config) # map alias flexmap oob-copy from iN1 dir a-to-b to it1 tag as-inline**
param traffic control	Specifies an option to pass GTP control traffic (GTP-c) to all GigaSMART engines in a GTP engine group. A GTP engine group has multiple GigaSMART engine port members. For example: **(config) # map alias to_vp_ctrl param traffic control** Refer to *"GTP Engine Grouping"* section in the *GigaVUE Fabric Management Guide* for details. Also refer to the *"GigaSMART GTP Correlation"* and the "*GigaSMART GTP Whitelisting and GTP Flow Sampling" section*s in the *GigaVUE Fabric Management Guide*.
priority <after <map name> | before <map name> | highest | lowest>	Sets the priority of the map relative to other maps. A packet matching multiple maps is sent to the map with the highest priority. For example: **(config) # map alias map1 priority before map2**
roles <assign | replace> <role> [to <role list>]	Assigns a user role to a map access list or replaces a map access list. For example: **(config) # map alias map1 roles assign monitor to listen_roles**

Argument	Description
rule add <drop | pass> <criteria>	Adds map rules (drop or pass), as follows: **add**—Creates a new pass rule. **drop**—Creates a new drop rule. Packets matching **drop** rules are dropped immediately without being sent to any configured shared collector or compared to any pass rules. Within a map, drop rules have precedence over pass rules. So, if a packet matches both a pass and a drop rule in the same map, the packet is dropped rather than passed. Both **pass** and **drop** rules have a wide variety of packet-matching criteria available, including MAC/IP addresses, application ports, VLAN IDs, and so on. Refer to map rule for rule criteria details. For example: **(config) # map alias map1** **(config map alias map1) # from 1/1/q1** **(config map alias map1) # to 1/1/q2** **(config map alias map1) # rule add pass vlan 100 comment "comment for rule"** **(config map alias map1) # comment "comment for whole template"** **(config map alias map1) # exit**
rule copy-from template <template alias>	Copies map rules from a template to create a map. • If there is a comment associated with the rule, it will be copied as well. • If there is a comment associated with the template as a whole, it will not be copied. For example, **(config) # map alias map1** **(config map alias map1) # from 1/1/q1** **(config map alias map1) # to 1/1/q2** **(config map alias map1) # rule copy-from template my_rule_template** **(config map alias map1) # exit**
rule delete <all | rule-id <rule ID>>	Deletes map rules, as follows: • **all** rules in the map. • a specified map rule in a map by **rule-id**. You can obtain the rule ID using the following command and typing the question mark (?) after the **rule-id** keyword. For example: **(config) # map alias add_header_1 rule delete rule-id ?<Integer> Rule Id13101112** You can also obtain the rule ID using the following

Argument	Description
	command and pressing the Tab key after the **rule-id** keyword. For example: **(config) # map alias add_header_1 rule delete rule-id1 3 10 11 12** • To delete a single rule: **(config) # map alias add_header_1 rule delete rule-id 1** • To delete multiple rules, separate them with commas as follows: **(config) # map alias add_header_1 rule delete rule-id 1,3,10** • To delete a range of rules, use the following syntax: **(config) # map alias add_header_1 rule delete rule-id 10..12** • To delete multiple rules including ranges, use the following syntax: **(config) # map alias add_header_1 rule delete rule-id 1,3,10..12** You can also obtain the rule ID for a specified map rule with the **show map alias <alias>** command.
rule edit rule-id <rule ID> <comment <comment> | drop <criteria> | pass <criteria>>	Edits a specified map rule in a map by **rule-id**. You can obtain the rule ID using the following command and typing the question mark (?) after the **rule-id** keyword. For example: **(config) # map alias dedup_1 rule edit rule-id ?<Integer> Rule Id12** You can also obtain the rule ID using the following command and pressing the Tab key after the **rule-id** keyword. For example: **(config) # map alias dedup_1 rule edit rule-id1 2** Once you have the **rule-id**, the following can be edited: • **comment <comment>**—Edits a map rule comment. • **drop <criteria>**—Edits the specified criteria in a drop rule. • **pass <criteria>**—Edits the specified criteria in a pass rule. Refer to map rule for rule criteria details. Maps with a subtype of ol, for overlap, such as flowSample-ol or flowWhitelist-ol, do not support map editing.
	For flexible inline arrangements, configures an external VLAN tag for a flexible inline map, as follows: • **1-4000**—Specifies a user-defined value for the

Argument	Description
	external VLAN ID in the range of 1 to 4000. • **auto**—Automatically assigns an external VLAN ID for packets going to inline tools. • The default is **auto**. The tag value is unique to each flexible inline map. For example: **(config) # map alias flexmap1 tag 100**
rewrite-dstmac <value> | rewrite-srcmac<value> **no rewrite-dstmac | no rewrite-srcmac**	For MAC Address rewrite ,configure the destination and Source fields as follows: • **rewrite-dstmac xx:xx:xx:xx:xx:xx** — Configure destination MAC rewrite for all the pass rules associated with the map. • **rewrite-srcmac xx:xx:xx:xx:xx:xx**— Configure source MAC rewrite for all the pass rules associated with the map. To unconfigure MAC Address rewrite for map based configuration use the below: • **no rewrite-dstmac** — Unconfigure the map level configured destination MAC rewrite value. • **no rewrite-srcmac** — Unconfigure the map level configured source MAC rewrite value. To delete a rule based MAC address re-write utilize the rule edit or delete command.
rewrite-dstip <value> | rewrite-srcip<value> **no rewrite-dstip | no rewrite-srcip vlan-op add <vlan id> tpid <value>**	For IP Address rewrite, configure the destination and Source fields as follows: • **rewrite-dstip x.x.x.x** — Configure destination IP rewrite for all the pass rules associated with the map. • **rewrite-srcip x.x.x.x**— Configure source IP rewrite for all the pass rules associated with the map. **NOTE:** The IP addresses 0.0.0.0, 255.255.255.255, and multicast are not accepted. To unconfigure IP Address rewrite for map based configuration use the below: • **no rewrite-dstip** — Unconfigure the map level configured destination IP rewrite value. • **no rewrite-srcip** — Unconfigure the map level configured source IP rewrite value. To delete a rule based IP address re-write utilize the rule edit or delete command.
vlan-op add <vlan id> tpid <value>	For VLAN manipulation, configure a VLAN ID and TPID value as follows: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095.

Argument	Description
	• **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100 and other supported values are 0x9100 and 0x88a8. To unconfigure VLAN manipulation for map-based configuration use the below: • **no vlan-op** — Unconfigure the map level configured VLAN tag.
vlan-op modify <vlan id> tpid <value>	For VLAN manipulation, modify a VLAN ID and TPID value as follows: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095. • **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100 and other supported values are 0x9100 and 0x88a8. To unconfigure VLAN manipulation for map-based configuration use the below: • **no vlan-op** — Unconfigure the map level configured VLAN tag.
vlan-op strip	It is used to remove or strip the outer VLAN tag from the incoming packet.
to <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias | one-arm| inline-tool-group-alias | inline-serial-alias | bypass | vport-alias | null-port>	Specifies the destination(s) for packets matching this map. Use one of the following: • **port-id, port-alias, port-list**—Sends matching traffic to one or more tool ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **gigastream-alias, gigastream-alias-list**—Sends matching traffic to the specified tool GigaStream. Refer to the "*GigaStream"* section in the *GigaVUE Fabric Management Guide* for details on GigaStream. • **inline-tool-alias**—Sends matching traffic to the specified inline tool alias. • **one-arm**- Configures one -arm mode for second level map. • **inline-tool-group-alias**—Sends matching traffic to the specified inline tool group alias. • **inline-serial-alias**—Sends matching traffic to the specified inline tool series alias. • **bypass**—Sends matching traffic to the specified inline bypass. • **vport-alias**—Sends matching GigaSMART traffic to the virtual port associated with the GigaSMART group. • **null-port**—Drops traffic after the GigaSMART operation is performed on the traffic. This is

Argument	Description
	applicable for regular maps and second-level maps. Refer to the "*Associating Inline Networks with Inline Tools Using Inline Maps*" section in the *GigaVUE Fabric Management Guide* for details on **inline-tool-alias**, **inline-tool-group-alias**, **inline-serial-alias**, and **bypass**. For example: **(config) # map alias map1 to 2/1/x1**
encap-tunnel <tunnel-alias>	Attaches the tunnel created for encapsulating the traffic. For example: **(config map alias <map-name>) # encap-tunnel <tunnel-alias>** To attach an encap-tunnel, ensure that you configure at least one circuit port in the **to** parameter.
type <firstLevel | flexInline | inline | regular | | transitLevel| secondLevel> firstLevel [byRule] flexInline [byRule | collector] inline [byRule] regular [byRule] secondLevel [byRule | flowFilter | flowSample | flowSample-ol | flowSample-sip | flowWhitelist | flowWhitelist-ol | flowWhitelist-sip]	Specifies the map type, as follows: • **regular**—Specifies a regular map type, with the **from** parameter specifying network or hybrid ports, or single inline-network or single inline-tool ports (for out-of-band maps) and the **to** parameter specifying tool or hybrid ports, GigaStream, or port group. • **inline**—Specifies an inline map type, with the **from** parameter specifying inline-network pairs or inline-network-groups and the **to** parameter specifying inline-tool pairs, inline-tool-group, inline-serial, or bypass. • **flexInline**—Specifies a flexible inline map type, which can only be applied to a single inline network. Each flexible inline map has its own VLAN ID. • **firstLevel**—Specifies a first level map type, with the **from** parameter specifying network or hybrid ports and the **to** parameter specifying virtual ports, used with GigaSMART operations. Specify the **firstLevel** map type when using the map **rule** parameter. • **transitLevel**—Specifies a transit level map type, with the **from** parameter specifying virtual ports, used with GigaSMART operations, and the **to** parameter specifying virtual ports. Specify the **transitLevel** map type when using a **gsrule** map rule. • **secondLevel**—Specifies a second level map type, with the **from** parameter specifying virtual ports, used with GigaSMART operations, and the **to** parameter specifying tool or hybrid ports, GigaStream, or port group. Specify the **secondLevel** map type when using a **gsrule**, **flowrule**, **flowsample**, or **whitelist** map rule. Also specifies the optional map subtype, as follows:

Argument	Description

	• **byRule**—Specifies a rule-based map subtype, which is supported on the following:
	  o **firstLevel**, **inline**, **flexInline**, and **regular** map types when using the map **rule** parameter.
	  o **secondLevel** map type when using the **gsrule** parameter.
	• **collector**—Specifies a collector map subtype. A collector map for flexible inline arrangements is defined as a subtype of flexible inline map. To create map passalls for flexible inline arrangements, you can define a collector map without any other maps.
	• **flowFilter**—Specifies a flow filtering map subtype, which applies to **secondLevel** map types. Specify the **flowFilter** map subtype when using a **flowrule** parameter.
	• **flowSample**—Specifies a flow sampling map subtype, which applies to **secondLevel** map types. Specify the **flowSample** map subtype when using a **flowsample** rule.
	• **flowSample-ol**—Specifies a flow sampling overlap map subtype, which applies to **secondLevel** map types. Specify the **flowSample-ol** map subtype when using a **flowsample** rule.
	• **flowSample-sip**—Specifies a SIP flow sampling map subtype, which applies to **secondLevel** map types.
	• **flowWhitelist**—Specifies a forward list map subtype, which applies to **secondLevel** map types. Specify the **flowWhitelist** map subtype when using a **whitelist** rule.
	• **flowWhitelist-ol**—Specifies a forward list overlap map subtype, which applies to **secondLevel** map types. Specify the **flowWhitelist-ol** map subtype when using a **whitelist** rule.
	• The default map subtype is **byRule**.
	For example:
	(config) # map alias map1 type inline byRule

use gsop <gsop alias>	Includes a named GigaSMART operation as part of this map, applying the associated GigaSMART functionality to packets matching any rule in the map (for example, slicing, de-duplication, Header Stripping, and so on). **NOTE: IMPORTANT—** GigaSMART operations **must** be added to the map before destination ports (**to**). This option is only available on nodes or clusters with GigaSMART features available and licensed. Refer to the "*Working with GigaSMART Operations*" *section* in the *GigaVUE Fabric Management Guide* for

Argument	Description
	details on creating GigaSMART operations. For example: **(config) # map alias map1 use gsop gsfilter**
whitelist add gtp <apn <pattern> | interface <Gn | S10 | S11 | S5> | version <1 | 2> | type <imsi | ran |all | wl-alias <alias name> >> delete all **add 5g<dnn<pattern> | type <supi | ran |all | wl-alias <alias name>> delete all**	Adds or deletes a rule in a forward list map as follows: • **add gtp**—Specifies adding a rule (a pass rule) to a forward list map. • **apn**—Specifies an Access Point Name (APN). • **interface**—Specifies a rule based on an Evolved Packet Core (EPC) interface. • **version**—Specifies a rule based on a GTP version. • **type**—Specifies the pattern required for the forward list DataBase (DB) lookup. o **imsi/supi** — Only IMSI or SUPI value used for the DB lookup. o **ran**— Only RAN value used for the DB lookup. o **all** — Both RAN and IMSI/SUPI value used for the DB lookup. **NOTE:** By default, SUPI or IMSI is value is used for the DB lookup, if no type is configured. • **wl-alias**—Specifies a maximum of ten forward list aliases in a single forward list map. The DB lookup happens only in the configured forward list alias based on the configured DB type. You must consider the following while configuring the forward list aliases: o When only DB type is configured and there is no forward list alias configured, then the first forward list DB configured in the gsparams is used for the DB lookup. o When there is no DB type and no whitleist alias are configured, then the lookup happens in all the forward list DB configured in the gsparams. • **delete all**—Specifies deleting the rules in an existing forward list map. To specify an APN, use a pattern, with or without a wildcard prefix or suffix, up to a maximum of 100 case-insensitive characters, as well as period (.), hyphen (-), and wildcard (*). APN is not supported on GigaVUE-HB1. To specify version, use the following: • 1 for v1 • 2 for v2 To specify EPC interfaces, use the following: • Gn for Gn/Gp • S11 for S11/S1-U

Argument	Description
	• S5 for S5/S8 • S10 **NOTE:** • Each forward list map can contain only one rule, with either a GTP version or an EPC interface. The rule can also specify an APN. • The rule cannot be edited. To edit a rule, first delete it, then recreate it. • GTP version and EPC interface are mutually exclusive. • A mix of versions and interface types across forward list maps, associated with the same vport, is not supported. This means you can have a maximum of two forward list maps with each map specifying a rule for version 1 and a rule for version 2, OR a maximum of four forward list maps with each map specifying a rule for each interface type. • Up to ten (10) forward list maps can be configured per vport. • Each forward list map, associated with the same vport, uses the same underlying forward list. For example: **(config) # map alias map1 whitelist add gtp version 1** **(config) # map alias map2 whitelist add gtp interface S5** Note that in the examples above, map1 and map2 would have to be associated with two different gsgroups. Other examples: **(config) # map alias map3 whitelist add gtp apn *mobile.com*** **(config) # map alias map1 whitelist delete all** Refer to the *"GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide* for more information.
whitelist add sip type <all | callee-id | caller-id | dest-ip | id | ip-addr | src-ip>	Adds a rule in a forward list map as follows: • **all**—Specifies adding a rule (a pass rule) to a forward list map based on caller, callee, source, desination or IP address. • **callee-id**—Specifies adding a rule (a pass rule) to a forward list map based on callee-id. • **caller-id**—Specifies adding a rule (a pass rule) to a forward list map based on caller-id.

Argument	Description
	• **dest-ip**—Specifies adding a rule (a pass rule) to a forward list map based on destination IP address. • **id**—Specifies adding a rule (a pass rule) to a forward list map based on callee or caller id. • **ip-addr**—Specifies adding a rule (a pass rule) to a forward list map based on destination IP address. • **src-ip**—Specifies adding a rule (a pass rule) to a forward list map based on source IP address. For example: **config) # map alias sip-whitelist-map add sip type ip-addr** Refer to the *"GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide* for more information.
map priority <map names>	Reorder map priority on an existing chain of maps. For example: **(config) # map priority map1**

Related Commands

The following table summarizes other commands related to the **map** command:

Task	Command
Displays all maps.	**# show map**
Displays map accessibility.	**# show map access**
Displays detailed information for a specified map, including its mapping and rules. **NOTE:** The existing 'show map' command is enhanced to display the 'SVT mode' as enabled only when Single Tag mode is enabled on Flexinline-SSL maps.	**# show map alias map1**
Displays detailed information on all maps.	**# show map all**
Displays map assignment.	**# show map assignment**
Displays map assignment for a specified map.	**# show map assignment alias map1**
Displays all maps in a table format.	**# show map brief**
Displays map mode.	**# show map mode**
Displays priority of all maps.	**# show map priority**
Displays priority of a specified map.	**# show map priority alias map1**
Displays statistics for a specified map.	**# show map stats alias map1**
Displays statistics for a specified rule.	**# show map stats**

Task	Command
	alias map1 rule 2
Displays all map counters.	**# show map stats all**
Displays all flexible inline maps.	**# show map-flexinline**
Displays detailed information for a specified flexible inline map.	**# show map-flexinline alias FLEX1**
Displays all flexible inline maps.	**# show map-flexinline all**
Deletes a specified map. **NOTE:** When you delete auto-generated maps (created while deploying solutions through GigaVUE-FM) via CLI, a warning message will be displayed along with a confirmation about the impact on solutions deployed in GigaVUE-FM.	**(config) # no map alias mymap**
Deletes the comments for a specified map.	**(config) # no map alias mymap comment**
Disables the specified map.	**(config) # no map alias mymap enable**
Modifies sources configured for a specified map. The delete must be followed immediately by the new **from** configuration.	**(config) # no map alias mymap from** **(config) # map alias mymap from 1/1/x1**
Deletes the option to pass traffic to tool if there is no matching rule.	**(config) # no map alias mymap no-rule-match pass**
Deletes the option to pass GTP control traffic (GTP-c) to all GigaSMART engines in a GTP engine group.	**(config) # no map alias mymap param traffic control**
Deletes an assigned role from a specified map.	**(config) # no map alias mymap roles assign monitor**
Deletes all assigned roles from a specified map.	**(config) # no map alias mymap roles assign all**
Deletes all destinations configured for a specified map.	**(config) # no map alias mymap to**
Deletes the GigaSMART operation associated with a specified map.	**(config) # no map alias mymap use gsop**
Deletes all maps.	**(config) # no map all**

Task	Command
Deletes the configured destination and source MAC Address.	**(config)# no rewrite-dstmac | no rewrite-srcmac**
Deletes the configured destination and source IP Address.	**(config)# no rewrite-dstip | no rewrite-srcip**
Deletes the configured VLAN manipulation from a specified map.	**(config)# no vlan-op**

logging

Required Command-Line Mode = Configure

Use the **logging** command to configure how the GigaVUE HC Series node stores syslog information—how much is stored, how the log files are handled, and so on.

NOTE: This section lists and describes the arguments for the **logging** command.

The **logging** command has the following syntax:

```
logging <hostname, IPv4 or IPv6 address> [tcp <0-65535> [ssh username <username>]] |
      [trap <severity level>]
  files
    delete <current | oldest [number of log files]>
    rotation force | max-num <number of files>
    upload <current | <file number>> <upload URL>
  level
    audit mgmt <severity level>
    cli commands <severity level>
    port <severity level>
  local <severity level>
  process <process> <severity level>
  trap <severity level>
```

The following table describes the arguments for the **logging** command:

Argument	Description
<hostname, IPv4 or IPv6 address>	Specifies the IP address for logging. Logged events are always written to the local log file. In addition, you can optionally specify an external syslog server as a destination for the GigaVUE HC Series node's logging output. When an external syslog server is specified, the GigaVUE HC Series node will send logged events through UDP to the specified destination. Use the **logging** command to specify an external syslog server. For example, the following command adds an IPv4 destination for syslog output: **(config) # logging 192.168.1.25** IPv6 addresses as well as hostnames are supported for logging. For example, the following command adds an IPv6 destination for syslog output: **(config) # logging 2001:db8:a0b:12f0::85** For example, the following command specifies a previously defined hostname: **(config) # logging syslog.ipv6** Refer to the following commands to configure a hostname for IPv4 or IPv6: ip or ipv6. By default, event logs will be sent to the syslog server using UDP.
[tcp <0-65535> [ssh username <username>]]	Specifies the TCP protocol. Syslog audit data will be sent to this server using TCP. A TCP port number must be specified, from 1 to 65535. The TCP port number is the port on which the syslog server listens. (Refer to your syslog server administrator for the port number.) A TCP port number of zero (0) specifies UDP. **NOTE:** TCP port numbers from 0 to 1024 are reserved for system use. For example, the following commands specify TCP ports, for IPv4 address, IPv6 address, and hostname: **(config) # logging 192.168.1.25 tcp 51300** **(config) # logging 2001:db8:a0b:12f0::85 tcp 1234** **(config) # logging syslog.ipv6 tcp 1468** Optionally specifies a secured TCP connection, which sends syslog audit data encrypted through SSH. The values are as follows: • **ssh**—Specifies that syslog audit data will be sent to this server using secured TCP. • **username**—Specifies a valid username for the secured TCP connection. This is the user account used for SSH authentication. Refer to **ssh connect** under ssh to configure a valid username. For example: **(config) # logging 192.168.1.25 tcp 51300 ssh username sysloguser**
[trap <severity level>]	Specifies an optional trap argument for the minimum severity for events and CLI commands sent to the specified remote destination. The value you specify here overrides the global setting configured using logging trap <severity level>. For example, the following command specifies a minimum severity level of critical for events sent to IPv4 address, 192.168.1.25:

Argument	Description
	(config) # logging 192.168.1.25 trap crit For example, the following commands specify a minimum severity level of information for events sent to an IPv6 hostname or IPv6 address: **(config) # logging syslog.ipv6 trap info** **(config) # logging 2001:db8:a0b:12f0::85 trap info**
files delete <current | all | oldest [number]> rotation force upload <current | <file number>> <upload URL>	Deletes log files, configures the rotation of log files, and uploads log files to an external host as follows: • **delete**—Deletes a log file. You can delete either the **current** log file or a specified number of the oldest log files using the **oldest** argument. For example, the following command deletes the three oldest log files: **(config) # logging files delete oldest 3** • **rotation force**—Forces the rotation of log files immediately. For example: **(config) # logging files rotation force** • **upload**—Uploads log files to an external host. Use **show log files** to see the list of files available for upload. Alternatively, you can use the **current** argument to upload the messages in the active log file or the **all** argument to upload all files. Use FTP, TFTP, or SCP to upload the file. The format for the upload URL is as follows: **[protocol]://username[:password]@hostname/path/filename** For example, the following command uploads the current log file to the FTP server at 192.168.1.25: **(config) # logging files upload current ftp://jhendrix:if6was9@192.168.1.25** Uploaded log files are stored in gzip format with a filename in the following format: **messages.<n>.gz.** Use the **masked** argument to mask the IPv4, IPv6, and MAC addresses contained in the log files. For example: **(config) # logging files upload all scp masked**

Argument	Description
level audit mgmt <severity level> cli commands <severity level> port <severity level>	Specifies the minimum severity for a CLI command to be logged to the local and remote syslogs. It also specifies the minimum severity of audit log messages. It also specifies the minimum severity of certain port log messages. Currently, port link up/down logs are supported. The available severity levels are listed in Severity Levels for Logging Commands. For example: **(config) # logging level cli commands info** **(config) # logging level audit mgmt notice** **(config) # logging level port alert**
local <severity level>	Specifies the minimum severity for an event to be logged to the local syslog. The available severity levels are listed in Severity Levels for Logging Commands. For example: **(config) # logging local crit**
trap <severity level>	Specifies the minimum severity for an event to be logged to the external syslog. The available severity levels are listed in Severity Levels for Logging Commands. For example: **(config) # logging trap alert**

Related Commands

The following table summarizes other commands related to the **logging** command:

Task	Command
Displays logging configuration.	**# show logging**
Displays the minimum severity of certain port log messages.	**# show logging port**
Does not send event logs to this server (removes the logging server configuration).	**(config) # no logging 10.10.10.10** or **2001:db8:a0b:12f0::85**
Disables logging using TCP, which reverts back to UDP. If there is an SSH connection configured under TCP, it will be unconfigured.	**(config) # no logging 10.10.10.10 tcp** or **(config) # logging 10.10.10.10 tcp 0**
Disables logging using a secured SSH connection.	**(config) # no logging 10.10.10.10 ssh**
Disables local logging.	**(config) # no logging local**
Does not send event log messages to syslog servers.	**(config) # no logging trap**

Severity Levels for Logging Commands

Use the following severity levels with the **logging local**, **logging trap**, and **logging level cli command** commands:

Log-Level	Description
emerg	Emergency—the system is unusable. The severity level with the least logging. Only emergency level events/commands are logged.
alert	Action must be taken immediately.
crit	Critical conditions.
err	Error conditions.
warning	Warning conditions.
notice	Normal but significant condition.
info	Informational messages.
debug	Debug-level messages. Authorized for factory use only.
none	Disable logging.

map rule

The **map rule** command has the following syntax:

```
rule add <drop | pass>
  bidir   comment <comment>
  circuit-id <2-4000>
  dscp <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef>
  ethertype <2-byte-hex>
  inner-ethertype <2-byte-hex>
  inner-ipver <4 or 6>
  inner-ipfrag <no-frag|all-frag|all-frag-no-first|first-frag|first-or-no-frag>
  inner-ipdst <IP address>
  inner-ipsrc <IP address>
  inner-ip6dst <IPv6 address>
  inner-ip6src <IPv6 address>
  inner-protocol <1-byte-hex>
  inner-portdst <0~65535 | x..y> inner-portdst-subset <even|odd>
  inner-portsrc <0~65535 | x..y> inner-portsrc-subset <even|odd>
  inner-uda1-data <16-byte-hex> inner-uda1-mask <16-byte-hex> inner-uda1-offset <0-120
bytes>
  inner-uda2-data <16-byte-hex> inner-uda2-mask <16-byte-hex> inner-uda2-offset <0-120
bytes>
  inner-vlan <vlan | vlan1..vlan2> innervlan-subset <even | odd>
  ip6dst <IPv6 address> <IPv6 netmask>
```

ip6fl <3-byte-hex>
ip6src <IPv6 address> <IPv6 netmask>
ipdst <IP address> <netmask>
ipfrag <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-frag>
ipsrc <IP address> <netmask>
ipver <4 | 6>
l2gre-id <1-4294967295>
macdst <MAC address> <MAC netmask>
macsrc <MAC address> <MAC netmask>
mpls-label-id <0~1048575 | x..y> pos <any | 1~7> mpls-label-id-subset <even|odd>
mpls-label-exp <0~7> pos <any | 1~7>
mpls-label-bos <0/1> pos <any | 1~7>
mpls-label-ttl <0~255 | x..y> pos <any | 1~7> mpls-label-ttl-subset <even|odd>
portdst <0-65535 | x..y> portdst-subset <even | odd>
portsrc <0-65535 | x..y> portsrc-subset <even | odd>
protocol <ipv6-hop | icmp-ipv4 | igmp | ipv4ov4 | tcp | udp | ipv6 | rsvp | gre | icmp-ipv6> <1-byte-hex>
rewrite-dstmac <value> rewrite-srcmac <value>
rewrite-dstip <value> rewrite-srcip <value>
vlan-op add <vlan id> tpid <value>
vlan-op modify <vlan id> tpid <value>
vlan-op strip
tcpctl <1-byte-hex> tcpctlmask <1-byte-hex>
tosval <1-byte-hex>
ttl <ttl | ttl1..ttl2>
uda1-data <16-byte-hex> uda1-mask <16-byte-hex> uda1-offset <2-110 bytes>
uda2-data <16-byte-hex> uda2-mask <16-byte-hex> uda2-offset <2-110 bytes>
vlan <vlan | vlan1..vlan2> vlan-subset <even | odd>
vxlan <1-16777215>

The following table describes the arguments for the **map rule** command:

Argument	Description
<drop | pass>	Adds a map drop rule or a map pass rule.
bidir	Mirrors source and destination rules on Layer 2-Layer 3 address and port number. The **bidir** argument automatically creates a second map rule mirroring source arguments to the destination (and vice-versa). For example, consider the following map rule: **(config) # map alias map1 rule add pass ipdst 192.168.1.50 255.255.255.0 ipsrc 192.168.1.25 255.255.255.0 bidir** Because the **bidir** argument is included, the system automatically creates a second map rule mirroring all source/destination criteria: **rule add pass ipsrc 192.168.1.50 255.255.255.0 ipdst 192.168.1.25 255.255.255.0** **NOTE:** The second map rule is not displayed in the output of the **show map** command,

Argument	Description
	however, it consumes map rule resources. You can also include the **bidir** argument with an IP source and port source as follows: **(config) # map alias map2 rule add pass ipsrc 192.168.1.22 /32 portsrc 23 bidir** The **bidir** argument causes the following rule to be added automatically: **rule add pass ipdst 192.168.1.22 /32 portdst 23** You can also include the **bidir** argument with just a single IP address to specify that you want to see traffic both in and out of a particular address. For example, this rule specifies that we want all traffic to and from 192.168.1.75: **(config) # map alias map3 rule add pass ipsrc 192.168.1.75 /32 bidir** The **bidir** argument causes the following rule to be added automatically: **rule add pass ipdst 192.168.1.75 /32** You can also include the **bidir** argument with an IP source and destination and a port source and destination as follows: **(config) # map alias map4 rule add pass ipsrc 192.168.1.33 /32 ipdst 192.168.1.44 /32 portsrc 23 portdst 63 bidir** The **bidir** argument causes the following rule to be added automatically: **rule add pass ipdst 192.168.1.33 /32 ipsrc 192.168.1.44 /32 portdst 23 portsrc 63** **NOTE:** If the bidir argument is added to a rule that does not support bidirectional filters, the bidir argument will not appear in the output of the show running-config command and there will not be any error message displayed. For example, the TCP protocol rule does not support bidirectional filters and the bidir argument does not appear in the output of the show running-config command for this rule.
comment <comment>	Add comments to map rules. Comments can be up to 128 characters, including special characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # map alias m1 rule add drop ipver 6 comment “Drop IPv6”**
circuit-id <2-4000>	Configure a user-defined Circuit ID within the range of 2 to 4000. For example: **(config) # map alias circuit_rule rule add pass circuit-id 2000**
dscp <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 |	Creates a map rule pattern for a particular decimal DSCP value. You can select any value within the four Assured Forwarding (af) class ranges or **ef** for **Expedited Forwarding** (the highest priority in the DSCP model). The valid DSCP values by Assured Forwarding Class are as follows: • **Class 1**—11, 12, 13 • **Class 2**—21, 22, 23 • **Class 3**—31, 32, 33 • **Class 4**—41, 42, 43 • **Expedited Forwarding**—ef

Argument	Description
af41 | af42 | af43 | ef>	For example, the following map rule passes all traffic with expedited forwarding assigned: **(config map alias mymap) # map alias m1 rule add pass dscp ef**
ethertype <2-byte-hex>	Creates a rule pattern for the ethertype value in a packet. For example, the following rule matches all traffic with an IPv6 ethertype (0x86DD): **(config map alias mymap) # rule add pass ethertype 0x86DD** **NOTE:** To create rules matching VLANs use the predefined VLAN map rule element type instead of the following TPID ethertypes: o 0x8100 o 0x88A8 o 0x9100 For details, refer to the "*Handling of Q-in-Q Packets in Map Rules*" section in the *GigaVUE Fabric Management Guide*. **NOTE:** The CLI accepts hexadecimal entries either with or without the leading **0x**.
inner-ethertype <2-byte-hex>	Creates a rule pattern for the inner-ethertype value in a packet. This inner qualifier is applicable only for GigaVUE-TA400. For example, the following rule matches all traffic with an IPv6 inner-ethertype (0x86DD): **(config map alias mymap) # rule add pass inner-ethertype 0x86DD** **NOTE:** The following TPID ethertypes are not supported: o 0x8100 o 0x88A8 o 0x9100 For details, refer to the "*Handling of Q-in-Q Packets in Map Rules*" section in the *GigaVUE Fabric Management Guide*. **NOTE:** The CLI accepts hexadecimal entries either with or without the leading **0x**.
inner-ipver <4 | 6>	Specifies the inner IP version for a map rule that matches either inner IPv4 or IPv6 traffic. This inner qualifier is applicable only for GigaVUE-TA400. Refer to the *"IPv4/IPv6 and Map Rules"* section in the *GigaVUE Fabric Management Guide* for more information on **inner-ipver**. For example: **(config map alias mymap) # rule add pass inner-ipver 4**
inner-ipfrag <no-frag | all-frag | all-frag-no-first |	Creates a map rule for different types of IPv4 and IPV6 inner fragments as follows: • **no-frag**—Matches inner unfragmented packets. • **all-frag**—Matches any inner fragment. • **all-frag-no-first**—Matches all inner fragments except the first inner fragment in a packet. • **first-frag**—Matches the first inner fragment of a packet.

Argument	Description
first-frag | first-or-no-frag>	• **first-or-no-frag**—Matches inner unfragmented packets or the first inner fragment of a packet. This inner qualifier is applicable only for GigaVUE-TA400. For example, **(config map alias mymap) # rule add pass inner-ipfrag first-frag** creates a rule that matches the first inner fragment in a packet.
inner-ipdst <IP address> **inner-ipsrc <IP address>**	Creates a rule for either an inner source IPv4 address or an inner destination IPv4 address. This inner qualifier is applicable only for GigaVUE-TA400. For example: **(config map alias mymap) # rule add pass inner-ipdst 20.0.0.1 /24** **(config map alias mymap) # rule add pass inner-ipsrc 30.0.0.1 255.255.0.0**
inner-ip6src <IPv6 address> **inner-ip6dst <IPv6 address>**	Creates a rule for either an inner source IPv6 address or an inner destination IPv6 address. This inner qualifier is applicable only for GigaVUE-TA400. Enter IPv6 addresses as eight 16-bit hexadecimal blocks separated by colons. For example: **2001:0db8:3c4d:0015:0000:0000:abcd:ef12** For example: **(config map alias mymap) # rule add pass inner-ip6dst fe80::01 /64** **(config map alias mymap) # rule add pass inner-ip6dst 1001:01::0a ffff:ffff:0000::0000**
inner-protocol **Inner protocol number 0** **Inner protocol number 1** **Inner protocol number 2** **Inner protocol number 4** **Inner protocol number 6** **Inner**	Creates a map rule for a particular inner protocol. This inner qualifier is applicable only for GigaVUE-TA400. For example, to create a map rule that excludes all inner GRE traffic, use the following **(config map alias gre-map) # rule add drop inner-protocol gre** **Protocol Map Rules and IPv6** The predefined inner protocol map-rules available for IPv4 (inner GRE, RSVP, and so on) are not allowed when **inner-ipver** is set to **6**. This is because, with the next header approach used by IPv6, the next layer of inner protocol data is not always at a fixed offset as it is in IPv4. To address this, the **<1-byte-hex>** option allows to match against the standard hex values for these inner protocols in the Next Header field. The standard 1-byte-hex values for both IPv4 and IPv6 are as follows: **0x00**: Hop-By-Hop Option (v6 only) **0x01**: ICMP (v4 only) **0x02**: IGMP **0x04**: IP over IP **0x06**: TCP **0x11**: UDP **0x29**: IPv6 over IPv4 **0x2b**: Routing Option (v6 only) **0x2c**: Fragment (v6 only)

Argument	Description
protocol number 17 **Inner protocol number 41** **Inner protocol number 46** **Inner protocol number 47** **Inner protocol number 58** **Custom hex entry**	**0x2E**: RSVP (v4 only) **0x2F**: Inner GRE (v4 only) **0x32**: Encapsulation Security Payload (ESP) Header **0x33**: Authentication (v6 only) **0x3a**: ICMP (v6 only) **0x3b**: No Next Header (v6 only) **0x3c**: Destination Option (v6 only)
inner-portdst <0-65535 | x..y> **inner-portdst-subset <even | odd>** **inner-portsrc <0-65535 | x..y>** **inner-portsrc-subset <even | odd>**	Creates a rule for an inner source application port or inner destination application port. You can specify the following: • A range of ports. For example, to match all inner source ports from 5000 to 5100, use the following: **(config map alias mymap) # rule add pass portsrc 5000..5100** • Either odd or even port numbers using the **inner-portdst-subset** and **inner-portsrc-subset** arguments. These arguments are useful when setting up rules for VoIP traffic. Most VoIP implementations send RTP traffic on even port numbers and RTCP traffic on odd port numbers. Following are some examples: • To match all odd inner source ports between 5000 and 5100, use the following: **(config map alias mymap) # rule add pass inner-portdst 5000..5100 inner-portdst-subset odd** • To match only the TCP/UDP traffic with the specific inner destination port, use the following: **(config map alias mymap)# rule add pass inner-protocol tcp inner-portdst 3000** **(config map alias mymap)# rule add pass inner-protocol udp inner-portdst 3000** • To match only the TCP/UDP traffic with the specific inner source port, use the following: **(config map alias mymap)# rule add pass inner-protocol tcp inner-portsrc 3000** **(config map alias mymap)# rule add pass inner-protocol udp inner-portsrc 3000**

Argument	Description
inner-uda1-data <16-byte-hex> inner-uda1-mask <16-byte-hex> inner-uda1-offset <0-120 bytes> inner-uda2-data <16-byte-hex>] inner-uda2-mask <16-byte-hex> inner-uda2-offset <0-120 bytes>	Creates up to two user-defined, 16-byte pattern matches in a rule. This inner qualifier is applicable only for GigaVUE-TA400. A pattern is a particular sequence of bits at a specified offset from the start of the inner L3 header. Inner user-defined pattern matches consist of the following: • A pattern (**inner-udax-data**). The pattern specifies what to search. The GigaVUE-TA400 supports 8-byte data match. The data to be matched must be specified with trailing 8-byte zeros • A mask (**inner-udax-mask**). The mask specifies the bits in the pattern that must match to satisfy the map rule. • An offset (**inner-udax-offset**). The offset specifies where in the packet the bits must match. A single rule can contain up to two user-defined pattern matches. **NOTE:** Always use the predefined map rule elements instead of user-defined pattern matches when possible. Examples: **(config map alias mymap) # rule add pass inner-uda1-data 26010101-2701010e-00000000-00000000 inner-uda1-mask ffffffff-ffffffff-00000000-00000000 inner-uda1-offset 12** **(config map alias mymap) # rule add pass inner-uda2-data 03ed0158-40060000-00000000-00000000 inner-uda2-mask ffffffff-ffff0000-00000000-00000000 inner-uda2-offset 4** Refer to the *"User-Defined Pattern Match on GigaVUE-TA400"* section in the *GigaVUE Fabric Management Guide* for supported combinations.
inner-vlan <vlan | vlan1..vlan2> innervlan-subset <even | odd>	Creates a rule for an inner VLAN ID or range of inner VLAN IDs, as follows: • **inner-vlan**—Specifies the VLAN ID value as a number between 1 and 4094 or VLAN ID range as <vlan1..vlan2>. • **innervlan-subset**—Specifies a subset of VLAN IDs to match, either even or odd VLAN IDs. Double-tagged packets have both an inner and an outer VLAN tag. The outer tag is detected when the ethertype is 0x8100, 0x88A8, or 0x9100. The inner tag is detected only when the ethertype is 0x8100. Examples: **(config map alias mymap) # rule add pass inner-vlan 100 innervlan-subset even** **(config) # map alias map1 rule add pass inner-vlan 100..200**
ip6src <IPv6 address> <IPv6	Creates a rule for either a source or destination IPv6 address or netmask. Enter IPv6 addresses as eight 16-bit hexadecimal blocks separated by colons. For example: **2001:0db8:3c4d:0015:0000:0000:abcd:ef12**

Argument	Description
netmask> **ip6dst <IPv6 address> <IPv6 netmask>**	Use netmask to match traffic from a range of IP addresses. You can enter netmasks either in 16-bit hexadecimal blocks separated by colons or in the bit count format (refer to "*Using Bit Count Netmasks*" section in the *GigaVUE Fabric Management Guide*). Note that netmasks used in IP map rules do not need to begin from the start of the address, nor do masked bits need to be contiguous. For example, theGigaVUE HC Seriesnode will accept a netmask where the masked bits start in the third octet, as follows—0.0.255.255. For example: **(config map alias mymap) # rule add pass ip6src FE80:0:0:0:202:B3FF:FE1E:8329 /64** **(config map alias mymap) # rule add pass ip6dst FE80:0000:0000:0000:0202:B3FF:FE1E:8329 FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** **NOTE:** When creating a by-rule map with IP filters through GigaVUE-OS CLI, an invalid netmask is accepted without error. For example: **HC3-3-R6 (config) # map alias m2HC3-3-R6 (config map alias m2) # from 1/1/g3** **HC3-3-R6 (config map alias m2) # rule add pass ipsrc 1.1.1.1 0.0.3.0** **HC3-3-R6 (config map alias m2) # exit** **HC3-3-R6 (config) #** An error should appear in this case, but it does not.
ip6fl <3-byte-hex>	Creates a rule for the 20-bit Flow Label field in an IPv6 packet. Packets with the same Flow Label, source address, and destination address are classified as belonging to the same flow. IPv6 networks can implement flow-based QoS using this approach. Specify the flow label as a 3-byte hexadecimal pattern. Note, however, that only the last 20 bits are used—the first four bits must be zeroes (specified as a single hexadecimal zero in the CLI). For example, to match all packets without flow labels, use the following map rule: **(config map alias mymap) # rule add pass ip6fl 0x000000** Alternatively, to match the flow label of 0x12345, use the following: **(config map alias mymap) # rule add pass ip6fl 0x12345** **NOTE:** The CLI accepts hexadecimal entries either with or without the leading **0x**.
ipfrag no-frag all-frag all-frag-no-first	Creates a map rule for different types of IPv4 and IPV6 fragments as follows: • **no-frag**—Matches unfragmented packets. • **all-frag**—Matches any fragment. • **all-frag-no-first**—Matches all fragments except the first fragment in a packet. • **first-frag**—Matches the first fragment of a packet. • **first-or-no-frag**—Matches unfragmented packets or the first fragment of a packet. For example, **(config map alias mymap) # rule ipfrag first-frag** creates a rule that matches the first fragment in a packet.

Argument	Description
first-frag first-or-no-frag	
ipdst <IP address> <netmask> **ipsrc <IP address> <netmask>**	Creates a rule for either a source or destination IPv4 address or netmask. Use netmask to match traffic from a range of IP addresses. You can enter netmasks using either dotted-quad notation (<xxx.xxx.xxx.xxx>) or in the bit count format (refer to *Using Bit Count Netmasks* on page 90). Note that netmasks used in IP rules do not need to begin from the start of the address, nor do masked bits need to be contiguous. For example, theGigaVUE HC Series node will accept a netmask where the masked bits start in the third octet, 0.0.255.255. For example: **(config map alias mymap) # rule add pass ipsrc 1.1.1.1 /32** **(config map alias mymap) # rule add pass ipdst 2.2.2.2 255.255.255.248**
ipver <4 | 6>	Specifies the IP version for a map rule that matches either IPv4 or IPv6 traffic. **NOTE:** The **ipver** argument is implicitly set to **4**. If you configure a rule without **ipver** specified, the GigaVUE HC Series node assumes that the IP version is 4. You can also set **ipver** to **6** and use it together with other arguments to change their meaning. Refer to the *"IPv4/IPv6 and Map Rules"* section in the *GigaVUE Fabric Management Guide* for more information on **ipver**. For example: **(config map alias mymap) # rule add pass ipver 4**
l2gre-id <1 to 4294967295>	Configure a user-defined L2GRE ID within the range of 1 to 4294967295. For example: **(config) # map alias circuit_rule rule add pass l2gre-id 1999**
macdst <MAC address> <MAC netmask> **macsrc <MAC address> <MAC netmask>**	Creates a rule for either a source or destination MAC address as follows: • Enter MAC addresses in the **01:23:45:67:89:AB** format, using colons to separate bytes. • Use a MAC netmask to define a range of MAC addresses that match the rule pattern. Enter netmasks in the **ffff.ffff.ffff** or **ff:ff:ff:ff:ff:ff** format. For example: **(config map alias mymap) # rule add pass macsrc 00:00:00:01:02:04 ff:ff:ff:ff:ff:ff** **(config map alias mymap) # rule add pass macsrc 0000.0001.0205 ffff.ffff.ffff** **(config map alias mymap) # rule add pass macdst 00:00:00:01:02:06 ff:ff:ff:ff:ff:ff** **(config map alias mymap) # rule add pass macdst 0000.0001.0207 ffff.ffff.ffff** Refer to How to Use MAC Address/Mask Map Rules for examples of how to use MAC address masks.
mpls-	Adds a rule to pass traffic based on MPLS label criteria. Specifies the **MPLS label ID** (0~1048575 or

Argument	Description
label-id <0~1048575 | x..y> pos <any | 1~7> mpls-label-id-subset <even|odd>	a range x..y), the **label position** (any or 1~7), and optionally filter by even or odd label IDs using **mpls-label-id-subset**. This MPLS header qualifier is applicable only for GigaVUE-TA400. For example: **(config map alias mymap) # rule add pass mpls-label-id 100..200 pos 2 mpls-label-id-subset even** **(config map alias mymap) # # rule add pass mpls-label-id 500 pos 4 mpls-label-id 600 pos 5** **(config map alias mymap) # rule add pass mpls-label-id 500 pos any** When using the MPLS header attributes (Label ID, EXP, BoS, and TTL) in the flow map rule configuration on the GigaVUE-TA400 chassis, setting these attributes to a value of 0 may lead to unintended packet matches. To ensure precise filtering, it is recommended to additionally filter based on the depth of the MPLS label using the BoS (Bottom of Stack) attribute. For example: If a user wants to filter packets with Label ID as 0 at position 2, adding the following rule will also match MPLS 1-labelled packets and non-MPLS packets that do not have an MPLS header at position 2: **(config map alias mymap) # rule add pass mpls-label-id 0 pos 2** To avoid incorrect matches, add the MPLS BoS attribute as well to filter at position 2: **(config map alias mymap) # rule add pass mpls-label-id 0 pos 2 mpls-label-bos 1 pos 2**
mpls-label-exp <0~7> pos <any | 1~7>	Adds a rule to pass traffic based on the MPLS label EXP (Experimental) field. Specifies the **EXP value** (0~7) and the **label position** (any or 1~7). This MPLS header qualifier is applicable only for GigaVUE-TA400. For example: **(config map alias mymap) # rule add pass mpls-label-exp 1 pos any** **(config map alias mymap) # rule add pass mpls-label-exp 2 pos 1** **(config map alias mymap) # rule add pass mpls-label-exp 3 pos 2** When using the MPLS header attributes (Label ID, EXP, BoS, and TTL) in the flow map rule configuration on the GigaVUE-TA400 chassis, setting these attributes to a value of 0 may lead to unintended packet matches. To ensure precise filtering, it is recommended to additionally filter based on the depth of the MPLS label using the BoS (Bottom of Stack) attribute. For example: If a user wants to filter packets with Label EXP as 0 at position 2, adding the following rule will also match MPLS 1-labelled packets and non-MPLS packets that do not have an MPLS header at position 2: **(config map alias mymap) # rule add pass mpls-label-exp 0 pos 2** To avoid incorrect matches, add the MPLS BoS attribute as well to filter at position 2: **(config map alias mymap) # rule add pass mpls-label-exp 0 pos 2 mpls-label-bos 1 pos 2**
mpls-label-bos	Adds a rule to pass traffic based on the MPLS label BOS (Bottom of Stack) bit. Specifies the **EXP value** (0~7) and the **label position** (any or 1~7). This MPLS header qualifier is applicable only for

Argument	Description
<0/1> pos <any | 1~7>	GigaVUE-TA400. For example: **(config map alias mymap) # rule add pass mpls-label-bos 1 pos any** **(config map alias mymap) # rule add pass mpls-label-bos 1 pos 1** **(config map alias mymap) # rule add pass mpls-label-bos 1 pos 2** When using the MPLS header attributes (Label ID, EXP, BoS, and TTL) in the flow map rule configuration on the GigaVUE-TA400 chassis, setting these attributes to a value of 0 may lead to unintended packet matches. To ensure precise filtering, it is recommended to additionally filter based on the depth of the MPLS label using the BoS (Bottom of Stack) attribute. For example: If a user wants to filter MPLS 2-labelled packets where Label ID 0 is not part of the outermost MPLS header, the following rule will also match MPLS 1-labelled packets and non-MPLS packets without an MPLS header at position 2: **(config map alias mymap) # rule add pass mpls-label-id 0 pos 2 mpls-label-bos 0 pos 1** To avoid incorrect matches, add the MPLS BoS attribute as well to filter at position 2: **(config map alias mymap) # rule add pass mpls-label-id 0 pos 2 mpls-label-bos 0 pos 1 mpls-label-bos 1 pos 2**
mpls-label-ttl <0~255 | x..y> pos <any | 1~7> mpls-label-ttl-subset <even|odd>	Adds a rule to allow (pass) MPLS packets based on their TTL (Time To Live) value and label position. The **mpls-label-ttl** parameter specifies the TTL value or range (**0–255** or **x..y**) to match, while the **pos** parameter defines the MPLS label position in the stack (**1–7** for specific positions or **any** for any position). The mpls-label-ttl-subset parameter further filters TTL values to only even or odd numbers. This MPLS header qualifier is applicable only for GigaVUE-TA400. For example: **(config map alias mymap) # rule add pass mpls-label-ttl 30..60 pos any mpls-label-ttl-subset odd** **(config map alias mymap) # rule add pass mpls-label-ttl 40 pos any** **(config map alias mymap) # rule add pass mpls-label-ttl 50 pos 1 mpls-label-ttl 70 pos 3** When using the MPLS header attributes (Label ID, EXP, BoS, and TTL) in the flow map rule configuration on the GigaVUE-TA400 chassis, setting these attributes to a value of 0 may lead to unintended packet matches. To ensure precise filtering, it is recommended to additionally filter based on the depth of the MPLS label using the BoS (Bottom of Stack) attribute. For example: If a user wants to filter packets with Label TTL as 0 at position 2, the following rule will also match MPLS 1-labelled packets and non-MPLS packets that do not have an MPLS header at position 2: **(config map alias mymap) # rule add pass mpls-label-ttl 0 pos 2** To avoid incorrect matches, add the MPLS BoS attribute as well to filter at position 2: **(config map alias mymap) # rule add pass mpls-label-ttl 0 pos 2 mpls-label-bos 1 pos 2**

Argument	Description
rewrite-dstmac <value> | rewrite-srcmac< value>	For MAC Address rewrite ,configure the destination and Source fields as follows: • **rewrite-dstmac xx:xx:xx:xx:xx:xx** — Configure destination MAC rewrite for the rules . • **rewrite-srcmac xx:xx:xx:xx:xx:xx**— Configure source MAC rewrite for the rules. **NOTE:** Rule based MAC rewrite feature is applicable only on pass rules To delete a rule based MAC address, re-write utilize the rule edit or delete command.
rewrite-dstip <value> | rewrite-srcip <value>	For IP Address rewrite, configure the destination and Source fields as follows: • **rewrite-dstip x.x.x.x** — Configure destination IP rewrite for the rules. • **rewrite-srcip x.x.x.x**— Configure source IP rewrite for the rules. **NOTE:** Rule based IP rewrite feature is applicable only on pass rules To delete a rule based IP address re-write utilize the rule edit or delete commands.
vlan-op add <vlan id> tpid <value>	For VLAN manipulation, configure a VLAN ID and TPID value: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095. • **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100, and other supported values are 0x9100 and 0x88a8.
vlan-op modify <vlan id> tpid <value>	For VLAN manipulation, modify a VLAN ID and TPID value as follows: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095. • **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100 and other supported values are 0x9100 and 0x88a8. **NOTE:** Rule based VLAN manipulation is applicable only on pass rules.
vlan-op strip	It is used to remove or strip the outer VLAN tag from the incoming packet.
portdst <0-65535 | x..y> portdst-subset <even | odd> **portsrc <0-65535 | x..y> portsrc-subset <even | odd>**	Creates a rule for a source or destination application port. You can specify the following: • A range of ports. For example, to match all source ports from 5000 to 5100, use the following: **(config map alias mymap) # rule add pass portsrc 5000..5100** • Either odd or even port numbers using the **portdst-subset** and **portsrc-subset** arguments. These arguments are useful when setting up rules for VoIP traffic. Most VoIP implementations send RTP traffic on even port numbers and RTCP traffic on odd port numbers. Following are some examples: • To match all odd source ports between 5000 and 5100, use the following: **(config map alias mymap) # rule add pass portdst 5000..5100 portdst-subset odd** • To match only the TCP/UDP traffic with the specific destination port, use the following: **(config map alias mymap)# rule add pass protocol tcp portdst 3000** **(config map alias mymap)# rule add pass protocol udp portdst 3000** • To match only the TCP/UDP traffic with the specific source port, use the following:

Argument	Description
	(config map alias mymap)# rule add pass protocol tcp portsrc 3000 **(config map alias mymap)# rule add pass protocol udp portsrc 3000** **NOTE:** For non-TCP/UDP packets, the portsrc matches the first and second bytes after the L3 header. Whereas, the portdst matches the third and fourth bytes after the L3 header.
protocol **Protocol number 0** **Protocol number 1** **Protocol number 2** **Protocol number 4** **Protocol number 6** **Protocol number 17** **Protocol number 41** **Protocol number 46** **Protocol number 47** **Protocol number 58** **Custom hex entry**	Creates a map rule for a particular protocol. For example, to create a map rule that excludes all GRE traffic, use the following **(config map alias gre-map) # rule add drop protocol gre** **Protocol Map Rules and IPv6** The predefined protocol map-rules available for IPv4 (GRE, RSVP, and so on) are not allowed when **ipver** is set to **6**. This is because with the next header approach used by IPv6, the next layer of protocol data is not always at a fixed offset as it is in IPv4. To address this, the GigaVUE HC Series node provides the **<1-byte-hex>** option to match against the standard hex values for these protocols in the Next Header field. The standard 1-byte-hex values for both IPv4 and IPv6 are as follows: **0x00**: Hop-By-Hop Option (v6 only) **0x01**: ICMP (v4 only) **0x02**: IGMP **0x04**: IP over IP **0x06**: TCP **0x11**: UDP **0x29**: IPv6 over IPv4 **0x2b**: Routing Option (v6 only) **0x2c**: Fragment (v6 only) **0x2E**: RSVP (v4 only) **0x2F**: GRE (v4 only) **0x32**: Encapsulation Security Payload (ESP) Header **0x33**: Authentication (v6 only) **0x3a**: ICMP (v6 only) **0x3b**: No Next Header (v6 only) **0x3c**: Destination Option (v6 only)
tcpctl <1-byte-hex> tcpctlmask <1-	Creates a one-byte pattern match map rule for the standard TCP control bits (URG, SYN, FIN, ACK, and so on). Use the **tcpctlmask** argument to specify which bits should be considered when matching packets. Refer to the Set Map Rules for TCP Control Bits section in the *GigaVUE Fabric Management*

Argument	Description
byte-hex>	*Guide* for a list of the hexadecimal patterns for each of the eight TCP flags, along with some examples. **NOTE:** Map rules using the **tcpctl** argument must also include the **protocol** argument set to 6 (TCP). For example: **(config map alias mymap) # rule add pass protocol 6 tcpctl 0x02**
tosval <1-byte-hex>	Creates a rule for the Type of Service (TOS) value in an IPv4 header. The TOS value is how some legacy IPv4 equipment implements quality of service traffic engineering. The standard values are: • Minimize-Delay: Hex **0x10** or **10** • Maximize-Throughput: Hex **0x08** or **08** • Maximize-Reliability: Hex **0x04** or **04** • Minimize-Cost: Hex **0x02** or **02** • Normal-Service: Hex **0000** or **00** **NOTE:** Most network equipment now uses DSCP to interpret the TOS byte instead of the IP precedence and TOS value fields. For example: **(config map alias mymap) # rule add pass tosval 0000**
ttl <ttl | ttl1..ttl2>	Creates a rule for the Time to Live (TTL—IPv4) or Hop Limit (IPv6) value in an IP packet, as a number between 0 and 255 as follows: • If there is no **ipver** argument included in the map rule (or if it is set to **4**), theGigaVUE HC Seriesnode matches the value against the TTL field in IPv4 packets. • If **ipver** is set to **6** in the map rule, the GigaVUE HC Series node matches the value against the Hop Limit field in IPv6 packets. The TTL and Hop Limit fields perform the same function, specifying the maximum number of hops a packet can cross before it reaches its destination. For example: **(config map alias mymap) # rule add pass ttl 0**
uda1-data <16-byte-hex> uda1-mask <16-byte-hex> uda1-offset <2-110 bytes>	Creates up to two user-defined, 16-byte pattern matches in a rule. A pattern is a particular sequence of bits at a specified offset from the start of a frame. User-defined pattern matches consist of the following: • A pattern (**udax-data**). The pattern specifies on what to search. • A mask (**udax-mask**). The mask specifies the bits in the pattern that must match to satisfy the map rule. • An offset (**udax-offset**). The offset specifies where in the packet the bits must match. A single rule can contain up to two user-defined pattern matches. **NOTE:** Always use the predefined map rule elements instead of user-defined pattern matches

<table>
<tr><th>Argument</th><th>Description</th></tr>
<tr><td>uda2-data <16-byte-hex>] uda2-mask <16-byte-hex> uda2-offset <2-110 bytes></td><td>
when possible.
The following table provides the packet format, hex value, and offset details for UDA-1 and UDA-2 across different platforms:
<table>
<tr><th rowspan="2">Platform</th><th colspan="3">UDA-1</th><th colspan="3">UDA-2</th></tr>
<tr><th>Packet Fromat</th><th>Hex-Value</th><th>Offset</th><th>Packet Fromat</th><th>Hex-Value</th><th>Offset</th></tr>
<tr><td>GigaVUE-TA400</td><td>Start of Outer Ethertype</td><td>8 bytes</td><td>0-160</td><td>Start of Outer IP header</td><td>8 bytes</td><td>0-160</td></tr>
<tr><td>GigaVUE-HC1P/GigaVUE-HCT/GigaVUE-TA25/GigaVUE-TA25E</td><td>Start of Outer Ethertype</td><td>12 bytes</td><td>16-116</td><td>Start of Packet</td><td>4 bytes</td><td>0-60</td></tr>
<tr><td>Other Platforms</td><td>Start of Packet</td><td>16 bytes</td><td>2-110</td><td>Start of Packet</td><td>16 bytes</td><td>2-110</td></tr>
</table>
Examples:
(config map alias mymap) # rule add pass uda1-data 26010101-2701010e-00000000-00000000 uda1-mask ffffffff-ffffffff-00000000-00000000 uda1-offset 12
(config map alias mymap) # rule add pass uda2-data 03ed0158-40060000-00000000-00000000 uda2-mask ffffffff-ffff0000-00000000-00000000 uda2-offset 4
Refer to the “Working with User-Defined Pattern Match Rules” section in the GigaVUE Fabric Management Guide for details.
</td></tr>
<tr><td>vlan <vlan | vlan1..vlan2> vlan-subset <even | odd></td><td>
Creates a rule for a VLAN ID or range of VLAN IDs for an outer VLAN tag. You can also use the optional vlan-subset argument to match even or odd VLAN IDs.
For example, to match all even VLAN IDs between 200 and 300, use the following:
(config map alias mymap) # rule add pass vlan 200..300 vlan-subset even
</td></tr>
<tr><td>vxlan-id <1 to 16777215></td><td>
Configure a user-defined L2GRE ID within the range of 1 to 16777215.
For example:
(config) # map alias circuit_rule rule add pass vxlan-id 500
</td></tr>
</table>

map gsrule

The criteria available for pass and drop GigaSMART rules (gsrule) used with Adaptive Packet Filtering in second level maps is as follows. Refer to the *"GigaSMART Adaptive Packet Filtering (APF)"* in the *GigaVUE Fabric Management Guide* for use cases and details on the feature as a whole.

The **map gsrule** command has the following syntax:

```
gsrule add <drop | pass>
  comment <comment>
  erspan id <range <erspanid1..erspanid2>> | <value <1-1024>>
  ethertype <any | pos <1-6>> <range <2-byte-hex..2-byte-hex> <subset <even | odd | none>> |
     <value <2-byte-hex>>
  gre key <range <4-byte-hex..4-byte-hex> <subset <even | odd | none>> | <value <4-byte-
hex>>
  gtp gtpu-teid <range <4-byte-hex..4-byte-hex> <subset <even | odd | none>> | <value <4-
byte-hex>>
  ipv4
    dscp <any | pos <1-3>> <value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 |
af42 |         af43>>
    dst <any | pos <1-3>> <range <ipv4_address..ipv4_address>> | <value <ipv4_address>
<netmask>>
    frag <any | pos <1-3>> <value <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-
frag>>
    protocol <any | pos <1-3>> <range <1-byte-hex..1-byte-hex> <subset <even | odd | none>> |
      <value <1-byte-hex..1-byte-hex>>
    src <any | pos <1-3>> <range <ipv4_address..ipv4_address>> | <value <ipv4_address>
<netmask>>
    tosval <any | pos <1-3>> <range <1-byte-hex..1-byte-hex>> | <value <1-byte-hex..1-byte-hex>>
    ttl <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0-255>>
  ipv6
    dscp <any | pos <1-3>> <value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 |
af42 |
      af43>>
    dst <any | pos <1-3>> <range <ipv6_address..ipv64_address>> | <value <ipv6_address>
<netmask>>
    flow-label <any | pos <1-3>> <range <3-byte-hex..3-byte-hex> <subset <even | odd | none>>
|        <value <3-byte-hex>>
    src <any | pos <1-3>> <range <ipv6_address..ipv6_address>> | <value <ipv6_address>
<netmask>>
  ipver <any | pos <1-3>> <value <4 | 6>>
  l4port
    dst <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0..65535>>
    src <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0..65535>>
  mac
    dst <any | pos <1-3>> <range <MAC_address..MAC_address>> | <value <MAC_address>
<netmask>>
    src <any | pos <1-3>> <range <MAC_address..MAC_address>> | <value <MAC_address>
<netmask>>
  mpls label <any | pos <1-4>> <range <label1..label2> <subset <even | odd | none>> | <value
    <0-1048576>>
```

pmatch <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>

pmatch RegEx <value> <offset | begin..end>

pmatch <mask <1 byte-hex> from <start-of-match <offset> | end-of-match <offset>> to <end-of-match <length> | end-of-packet | <length>> <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>

pmatch-hint <hint string>

tcp ctl <any | pos <1-3>> <value <1-byte-hex>> <mask <1-byte-hex | none>>

vlan id <any | pos <1-4>> <range <vlan1..vlan2>> <subset <even | odd | none>> | <value <0-4094>>

vntag

dvifid <any | pos <1-3>> <range <dvifid1..dvifid2>> <subset <even | odd | none>> | <value <0-16384>>

svifid <any | pos <1-3>> <range <svifid1..svifid2>> <subset <even | odd | none>> | <value <0-4096>>

viflistid <any | pos <1-3>> <range <viflistid1..viflistid2>> <subset <even | odd | none>> | <value <0-16384>>

vxlan id <range <3-byte-hex..3-byte-hex>> <subset <even | odd | none>> | <value <3-byte-hex>>

The following table describes the arguments for the **map gsrule** command:

Argument	Description
add <drop | pass>	Adds a map drop gsrule or a map pass gsrule.
comment <comment>	Adds comments to map rules. Comments can be up to 128 characters, including special characters. Comments longer than one word must be enclosed in double quotation marks. For example: **map alias 1 gsrule add drop any value 6 comment "Drop IPv6"**
erspan id <range <erspanid1..erspanid2> | value <1-1024>>	Specifies an ERSPAN ID as a decimal value from 1 to 1024. You can enter either a range or a single value. For example: **(config map alias m1) # gsrule add pass erspan id value 0**
ethertype <any | pos <1-6>> <range <2-byte-hex..2-byte-hex> <subset <even | odd | none>> | value <2-byte-hex>	Specifies an ethertype as a two-byte hex value (for example, 0x86DD will match all traffic with an IPv6 ethertype). You can enter either a range or a single value. The range can include a subset. **NOTE:** The **<subset <even | odd | none>** condition is not supported in Gen 3 nodes. For example: **(config map alias m1) # gsrule add pass ethertype**

Argument	Description
	any value 0x86DD You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
gre key <range <4-byte-hex..4-byte-hex> <subset <even | odd | none>> | value <4-byte-hex>>	Specifies a GRE key as a four-byte hex value.You can enter either a range or a single value. The range can include a subset.
gtp gtpu-teid <range <4-byte-hex..4-byte-hex> <subset <even | odd | none>> | value <4-byte-hex>>	Specifies a GTP tunnel identifier as a four-byte hex value. You can enter either a range or a single value. The range can include a subset. For example: **(config map alias m1) # gsrule add pass gtp gtpu-teid range 0x001e8480..0x001e8489 subset none**
ipv4 dscp <any | pos <1-3>> <value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43>>	Specifies an IPv4 DSCP value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv4 dst <any | pos <1-3>> <range <ipv4_address..ipv4_address>> | <value <ipv4_address> <netmask>>	Specifies a destination IPv4 address and mask in standard dotted-quad format. You can enter either a range or a single value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv4 frag <any | pos <1-3>> <value <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-frag>>	Specifies IPv4 fragments to match to follows: • **no-frag**—Matches unfragmented packets. • **first-frag**—Matches the first fragment of a packet. • **first-or-no-frag**—Matches unfragmented packets or the first fragment of a packet. • **all-frag-no-first**—Matches all fragments except the first fragment in a packet. • **all-frag**—Matches any fragment. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv4 protocol <any | pos <1-3>> <range <1-byte-hex..1-byte-hex> <subset <even | odd | none>> | <value <1-byte-hex..1-byte-hex>>	Specifies an IPv4 protocol identifier as a one-byte hex value. You can enter either a range or a single value. The range can include a subset. Common IPv4 protocol identifiers include the following: • 0x00: Hop-By-Hop Option (v6 only) • 0x01: ICMP (v4 only) • 0x02: IGMP • 0x04: IP over IP • 0x06: TCP • 0x11: UDP • 0x29: IPv6 over IPv4

Argument	Description
	• 0x2b: Routing Option (v6 only) • 0x2c: Fragment (v6 only) • 0x2E: RSVP (v4 only) • 0x2F: GRE (v4 only) • 0x32: Encapsulation Security Payload (ESP) Header (v6 only) • 0x33: Authentication (v6 only) • 0x3a: ICMP (v6 only) • 0x3b: No Next Header (v6 only) • 0x3c: Destination Option (v6 only)
ipv4 src <any | pos <1-3>> <range <ipv4_address..ipv4_address>> | <value <ipv4_address> <netmask>>	Specifies a source IPv4 address and mask in standard dotted-quad format. You can enter either a range or a single value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv4 tosval <any | pos <1-3>> <range <1-byte-hex..1-byte-hex>> | <value <1-byte-hex..1-byte-hex>>	Specifies an IPv4 ToS value Flow as a three-byte hex value. You can enter either a range or a single value. The TOS value is how some legacy IPv4 equipment implements quality of service traffic engineering. The standard values are as follows: • Minimize-Delay: Hex 0x10 or 10 • Maximize-Throughput: Hex 0x08 or 08 • Maximize-Reliability: Hex 0x04 or 04 • Minimize-Cost: Hex 0x02 or 02 • Normal-Service: Hex 0000 or 00 **NOTE:** Most network equipment now uses DSCP to interpret the TOS byte instead of the IP precedence and TOS value fields. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv4 ttl <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0-255>>	Specifies an IPv4 Time-to-Live value as a decimal value from 0 to 255. You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv6 dscp <any | pos <1-3>> <value <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43>>	Specifies an IPv6 DSCP value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv6 dst <any | pos <1-3>> <range <ipv6_address..ipv64_address>> | <value <ipv6_address> <netmask>>	Specifies a destination IPv6 address and mask in standard dotted-quad format. You can enter either a range or a single value. You can also specify the field position for the attribute. Refer to

Argument	Description
	Specifying Field Position for GSRule Criteria for details.
ipv6 flow-label <any | pos <1-3>> <range <3-byte-hex..3-byte-hex> <subset <even | odd | none>> | <value <3-byte-hex>>	Specifies an IPv6 Flow Label as a three-byte hex value. You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipv6 src <any | pos <1-3>> <range <ipv6_address..ipv6_address>> | <value <ipv6_address> <netmask>>	Specifies a source IPv6 address and mask in standard dotted-quad format. You can enter either a range or a single value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
ipver <any | pos <1-3>> <value <4 | 6>>	Specifies an IP version of either 4 or 6. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
l4port dst <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0..65535>> src <any | pos <1-3>> <range <x..y> <subset <even | odd | none>> | <value <0..65535>>	Specifies a source or destination IP port number as a decimal value from 0 to 65535. You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
mac dst <any | pos <1-3>> <range <MAC_address..MAC_address>> | <value <MAC_address> <netmask>> src <any | pos <1-3>> <range <MAC_address..MAC_address>> | <value <MAC_address> <netmask>>	Specifies a source or destination MAC address and mask in hex format. You can enter either a range or a single value. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
mpls label <any | pos <1-4>> <range <label1..label2> <subset <even | odd | none>> | <value <0-1048576>>	Specifies an MPLS label as a decimal value from 0 to 1048576. You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
pmatch <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>	Specifies a Perl-compatible regular expression to be used as a filter as follows: • Use the **string** to pass all packets including the string **www.gigamon.com:** **(config map alias m1) # gsrule add pass pmatch string “www.gigamon.com” 0..1750** • Use the **RegEx** to pass packets matching any phone number in the nnn-nnn-nnnn format: **(config map alias m1) # gsrule add pass pmatch RegEx "^\d{3}-\d{3}-\d{4}$" 0..1750**

Argument	Description
	• The **protocol** specifies that the matching will start after the protocol header (IPv4, IPv6, TCP, or UDP). Pos 1 or 2 indicates the position, for example, pos 2 indicates that matching is to start after the second protocol header. Examples: **(config map alias m1) # gsrule add pass pmatch protocol tcp pos 1 RegEx "\x16\x03.{3}\x01" 0** **(config map alias m2) # gsrule add drop pmatch protocol tcp pos 1 string “octet-stream” 0..1000** • The **offset** is a value or range from 0 to 1750. The offset indicates where the pattern under search is located. Specify a value to indicate that the pattern has to start at that offset in the packet in order to be considered a match. Specify a range to indicate that the pattern can be anywhere in the packet in that range. • The **begin..end** specifies the start and end value of the range. Note the following: • If a string is used in a rule (RegEx or string arguments) and the string has spaces, enclose it in double quotation marks. • The character limit is 127 for configuring the RegEx pattern. • To include a double quotation mark (") in a string (a double quotation mark inside a double quotation mark, precede the inner double quotation mark with a backslash (\). For example: **(config map alias m1) # gsrule add pass pmatch string "SOAPAction: \"http://tempuri.org/HelloWorld\"\r\n" 34** For details on pattern matching, refer to the “*Pattern Matching on Gen 2 GigaSMART modules*” section in the *GigaVUE Fabric Management Guide* for details.
pmatch RegEx <value> <offset | begin..end>	This command is specific to Gen 3 GigaSMART module and allows the use of a Perl-compatible regular expression (RegEx) to create a filter. • The **value** is the actual pattern or string to match within the packet. • The **offset | begin..end** In the Gen 3 GigaSMART module, pattern matching has been simplified for ease of configuration. Users can specify a start offset with 0 as the default and the system will automatically scan up to 1750 bytes within the packet payload. Pattern Matching begins from specific locations based on the protocol type. For Layer 3 protocols (IPv4 and IPv6) and Layer 4 protocols (TCP and UDP) the pattern matching starts after the last protocol header. For all other protocols it begins at the start of the packet.

Argument	Description
	• The differences between the Gen 2 and Gen 3 map configurations for Pattern Match RegEx are listed in the Gen 2 map and Gen 3 map configuration for Pattern Match RegEx section. • Use the **RegEx** to pass all packets including the string **www.gigamon.com:** **(config map alias m1) # gsrule add pass pmatch "www.gigamon.com" 0** • Use the **RegEx** to pass packets matching any phone number in the nnn-nnn-nnnn format: **(config map alias m1) # gsrule add pass pmatch RegEx "^\d{3}-\d{3}-\d{4}$" 0** Note the following: • If a string is used in a rule (RegEx argument) and the string has spaces, enclose it in double quotation marks. • The character limit is 127 for configuring the RegEx pattern. • To include a double quotation mark (") in a string (a double quotation mark inside a double quotation mark, precede the inner double quotation mark with a backslash (\). For example: **(config map alias m1) # gsrule add pass pmatch string "SOAPAction: \"http://tempuri.org/HelloWorld\"\r\n" 34** • If the matching string or pattern appears beyond the 1750-byte limit, it will not be detected, even if it exists within the packet payload. For details on pattern matching, refer to the "*Pattern Matching on Gen 3 GigaSMART modules*" section in the *GigaVUE Fabric Management Guide* for details.
pmatch <mask <1 byte-hex> from <start-of-match <offset> | end-of-match <offset>> to <end-of-match <length> | end-of-packet | <length>> <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>	Specifies masking with pattern matching. Use masking to mask out a specific portion of a packet due to security reasons or to hide sensitive information in packets. APF allows masking when there is a match through pattern matching (**pmatch**). The parameters are as follows: • **mask <1 byte-hex>**—specifies that the matched pattern in the **gsrule** will be masked with the pattern specified in the **1-byte** masking pattern. The pattern specified in the **gsrule** will be overwritten. The overwritten length will be the length specified in either the **string** or the **RegEx pmatch**. Use the **1-byte** to overwrite the original pmatch pattern. If there are multiple matches in the packet, up to 10 matches will be masked. For example, to find Social Security numbers (in the format xxx-xx-xxxx) between offset 40 and 80 and replace them with zeros: **(config map alias m1) # gsrule add pass pmatch**

Argument	Description
	mask 0x00 RegEx "\d{3}-?\d{2}-?\d{4}" 40..80 On Gen3 GigaSMART modules , APF Regex masking scans up to 1750 bytes of payload and will mask all matches found within that region. It is not bounded to 10 matches • **from**—specifies that the next parameter (either **start-of-match** or **end-of-match**) will be the position in the packet where masking will start once the pattern is found. • **start-of-match <offset>**—specifies that masking will start at the **offset** number of bytes after the beginning of the matching pattern. The offset is from 0 to 1749. • **end-of-match <offset>** — specifies that masking will start at the offset number of bytes after the end of the matching pattern. The offset is from 0 to 1748. • **to**—specifies that the next parameter (either **end-of-match** or **end-of-packet**) will be in the position in the packet where masking will stop • **end-of-match** —specifies that masking will stop (or start) at the **offset** number of bytes after the end of the matching pattern. The offset is from -1749 to 1749. • **end-of-packet**—specifies that masking will stop at the end of the packet. • **length**—specifies the number of bytes the packet will be masked, starting from the beginning (or the end) of the matching pattern. The length is from 1 to 1750. **NOTE:** For all masking modes, the combined range of offset and length must remain within the maximum supported window of 1750 bytes (measured after the IP/TCP/UDP header). Even if the offset and length values are individually within their allowed ranges, the configuration is invalid if their sum exceeds 1750. For example: Invalid: **gsrule add pass mask 0x0 from start-of-match 1749 to 2 protocol tcp pos 1 RegEx "\d{3}-?\d{2}-?\d{4}" 40..80** Here 1749 + 2 = 1751, h exceeds the 1750 byte window. Valid : **gsrule add pass mask 0x0 from start-of-match 1749 to 1 protocol tcp pos 1 RegEx "\d{3}-?\d{2}-?\d{4}" 40..80** Here 1749 + 1 = 1750, is within the supported range. When configuring masking with from start-of-match, the mask length can be set from 1 to 1750 bytes. When configuring masking with from end-of-match, the mask length can be set

Argument	Description
	from 1 to 1749 bytes. • Refer to details of **string**, **protocol**, **RegEx**, and **begin..end** listed above. The syntax of this command can be one of the following: **pmatch <mask <1 byte-hex> from start-of-match <offset> to <end-of-match <length> | end-of-packet | <length>> <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>** or **pmatch <mask <1 byte-hex> from end-of-match <offset> to <end-of-packet | <length>> <protocol <ipv4 | ipv6 | tcp | udp>> <pos <1 | 2>> <string <pattern> | RegEx> <pattern> <offset | begin..end>** Examples: In a Gen 2 GigaSMART module **(config map alias m1) # gsrule add pass mask 0x0 from start-of-match 0 to end-of-packet protocol tcp pos 1 RegEx "\d{3}-?\d{2}-?\d{4}" 40..80** **(config map alias m2) # gsrule add pass pmatch mask 0xff from end-of-match 0 to 13 protocol udp pos 1 string "user_id=" 0..30** **(config map alias m3) # gsrule add pass pmatch mask 0xbb from end-of-match 1 to 40 RegEx "(cardid=)|(cardid\"\:\")" 0..1750** In a Gen 3 GigaSMART module **(config map alias m1) # gsrule add pass mask 0x0 from start-of-match 0 to end-of-packet RegEx "\d{3}-?\d{2}-?\d{4}" 0** **(config map alias m2) # gsrule add pass pmatch mask 0xff from end-of-match 0 to 13 RegEx "user_id=" 0** For masking with pattern matching, refer to the *"Masking with Pattern Matching"* section in the *GigaVUE Fabric Management Guide*.
pmatch-hint <hint string>	Specifies a pattern matching hint for a gsrule in a second level map. Use the hint to optimize APF pattern matching performance. The recommended string length is from 3 to 6 characters. The maximum string length is 63 characters. For example: **(config map alias mymap) # gsrule add pass pmatch RegEx a[gG]igamon|aGIMO\\s[a-f]\\d{4} 0..1750 pmatch-hint "gamon|GIM"**

Argument	Description
	For details on the pattern matching hint, refer to the "*Pattern Matching Hint"* section in the *GigaVUE Fabric Management Guide*.
tcp ctl <any | pos <1-3>> <value <1-byte-hex>> <mask <1-byte-hex | none>>	Specifies a one-byte pattern match filter for the standard TCP control bits described in the *"Setting Map Rules for TCP Control Bits"* section (URG, SYN, FIN, ACK, and so on) in the GigaVUE Fabric Management Guide for details. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
vlan id <any | pos <1-4>> <range <vlan1..vlan2>> <subset <even | odd | none>> | <value <0-4094>>	Specifies a VLAN ID as a decimal value from 0 to 4094. You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
vntag dvifid <any | pos <1-3>> <range <dvifid1..dvifid2>> <subset <even | odd | none>> | <value <0-16384>> svifid <any | pos <1-3>> <range <svifid1..svifid2>> <subset <even | odd | none>> | <value <0-4096>> viflistid <any | pos <1-3>> <range <viflistid1..viflistid2>> <subset <even | odd | none>> | <value <0-16384>>	Specifies one of the following VNTAG options: • **dvifid**—Specifies the destination VNTAG VIF ID as a decimal value. Valid destination IDs range from 0 to 16384. • **svifid**—Specifies the source VNTAG VIF ID as a decimal value. Valid source IDs range from 0 to 4096. • **viflistid**—Specifies the VIF List ID as a two-byte hex value (for example, 0x86DD will match all traffic with an IPv6 ethertype). You can enter either a range or a single value. The range can include a subset. You can also specify the field position for the attribute. Refer to Specifying Field Position for GSRule Criteria for details.
vxlan id <range <3-byte-hex..3-byte-hex>> <subset <even | odd | none>> | <value <3-byte-hex>>	Specifies a VXLAN ID as a three-byte hex value.You can enter either a range or a single value. The range can include a subset.

Specifying Field Position for GSRule Criteria

Packets using tunneling, encapsulation, MPLS, or any other tagging/encapsulation implementations include multiple versions of the same type of protocol header (for example, multiple VLAN tags, multiple IP headers, and so on). Many of the GigaSMART rule criteria let you specify which occurrence of a particular attribute you want to match. In these cases, you can specify either **any** to match ANY occurrence, or you can specify which occurrence of each attribute to filter using the **pos <1-6>** argument.

Refer to the following table for a summary of the maximum value allowed in the field position (**pos** argument) for each attribute supported.

Attribute	Maximum Occurrences
Attributes in IPv4 header	3
Attributes in IPv6 header	3
Attributes in MAC header	3
VLAN ID	4
MPLS Label	4
Attributes in Layer 4 port (l4port)	3
Ethertype	6
Attributes in VNTAG header	3
Attributes in TCP header	3
IP Version	3

map-group

Required Command-Line Mode = Configure

Use the **map-group** command to create a group of maps for GTP forward listing and GTP flow sampling. All the maps in a map group receive traffic according to map rules, rather than map priority. Thus, multiple copies of a GTP packet can be sent to more than one tool. This functionality is referred to as overlapping maps.

On the virtual port, a mode of **gtp-overlap** must be specified. Refer to vport.

The **map-group** command has the following syntax:

```
map-group alias <alias>
  comment <comment>
  map-list <list of maps>
```

The following table describes the arguments for the **map-group** command:

Argument	Description
alias <alias>	Specifies the name of the map group. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # map-group alias mg1**
comment <comment>	Specifies a unique text string that describes the map group. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # map-group alias mg1 comment “Shared collector”**

Argument	Description
map-list <list of maps>	Specifies the maps in a list separated by commas. For example: **(config) # map-group alias mg1 map-list lev2_fs1_4,lev2_fs2_4,lev2_fs3_4** NOTES: • A map group can be associated with only one GigaSMART group (gsgroup). • All maps in the map list have to be associated with the same gsgroup. • All maps within a map group must be connected to the same vport. • A map group can consist of only one GTP forward listing map or only one GTP flow sampling map but it cannot contain two maps of the same type. • Once a map group is created, it cannot be edited to change the type or subtype of the map. However, you can add and edit the map rules for a map while it is configured in a map group. • If multiple map groups are configured, the maps within each map group must point to the same port groups as the other map groups.

Related Commands

The following table summarizes other commands related to the **map-group** command:

Task	Command
Displays all map groups.	**# show map-group**
Displays information for a specified map group.	**# show map-group alias mg1**
Displays all map groups.	**# show map-group all**
Displays all map groups in table format.	**# show map-group brief**
Deletes a specified map group.	**(config) # no map-group alias mg1**
Deletes all map groups.	**(config) # no map-group all**

map-passall

Required Command-Line Mode = Configure

Use the **map-passall** command to send all packets on a network port to one or more tool ports or tool GigaStream irrespective of the maps already in place for the ports. Refer to the section *"Working with Map-Passalls and Port Mirroring"* in the *GigaVUE Fabric Management Guide* for a discussion of use cases for map-passalls.

NOTE: You can only use map-passall connections between ports/GigaStream on the same node. Cross-node map-passall connections are not supported.

The **map-passall** command has the following syntax:

```
map-passall alias <alias>
  comment <comment>
from <port-id | port-alias | inline-network-alias | inline-network-group-alias>
  roles <assign | replace> <role> [to <role list>]
  to <tool port list | gigastream-alias | gigastream-alias-list | inline-tool-alias | inline-tool-
group-alias |
      inline-serial-alias | bypass>
```

The following table describes the arguments for the **map-passall** command:

Argument	Description
alias <alias>	Specifies the name of the map passall. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # map-passall alias map2** **(config map-passall alias map2) #**
comment <comment>	Supplies an optional comment for this map-passall. The comment will appear in **show map-passall** output. For example: **(config) # map-passall alias map2 comment "Map from A to B"**
from <port-id | port-alias | inline-network-alias | inline-network-group-alias>	Specifies the source(s) for packets matching this map-passall. Use one of the following: • **port-id, port-alias**—Sends matching traffic from one or more network ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **inline-network-alias**—Sends matching traffic from the specified inline network alias. • **inline-network-group-alias**—Sends matching traffic from the specified inline network group alias. Refer to the "*Associating Inline Networks with Inline Tools Using Inline Maps"* section in the *GigaVUE Fabric Management Guide* for details on **inline-network-alias** and **inline-network-group-alias**. For example: **(config) # map-passall alias map2 from port1**
roles <assign | replace> <role> [to <role list>]	Assigns a user role to a map access list or replaces a map access list. For example: **(config) # map-passall alias map2 roles replace monitor to view_roles**
to <tool port list | gigastream-alias |	Specifies the destination(s) for packets matching this map-

Argument	Description
gigastream-alias-list | inline-tool-alias | inline-tool-group-alias | inline-serial-alias | bypass>	passall. Use one of the following: • **tool-port-list**—Sends matching traffic to one or more tool ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **gigastream-alias, gigastream-alias-list**—Sends matching traffic to the specified tool GigaStream. Refer to the *"GigaStream"* section in the *GigaVUE Fabric Management Guide* for details on GigaStream. • **inline-tool-alias**—Sends matching traffic to the specified inline tool alias. • **inline-tool-group-alias**—Sends matching traffic to the specified inline tool group alias. • **inline-serial-alias**—Sends matching traffic to the specified inline tool series alias. • **bypass**—Sends matching traffic to the specified inline bypass. Refer to the "*Associating Inline Networks with Inline Tools Using Inline Maps"* section in the *GigaVUE Fabric Management Guide* for details on **inline-tool-alias**, **inline-tool-group-alias**, **inline-serial-alias**, and **bypass**. For example: **(config) # map-passall alias map2 to inTool**

The following table shows some examples of configuring **map-passalls**:

Command	Comments
Map Prefix Mode Technique **(config) # map-passall alias mypass** (config map-passall alias **mypass**) # from 1/1/x1..x4 (config map-passall alias **mypass**) # to **1/1/x5** (config map-passall alias **mypass**)# exit **Separate Commands Technique** (config) # map-passall alias **mypass** from 1/1/x1..x4 (config) # map-passall alias **mypass** to **1/1/x5**	Configures a map-passall from 1/1/x1, 1/1/x2, 1/1/x3, and 1/1/x4 to tool port **1/1/x5**.
Map Prefix Mode Technique **(config) # map-passall alias mypass2**	Configures a map passall from 1/2/x1 to 1/2/x2, 1/2/x3, 1/2/x4, and **1/2/x5**.

Command	Comments
(config map-passall alias **mypass2**) # from 1/2x1 (config map-passall alias **mypass2**) # to **1/2/x2..x5** (config map-passall alias **mypass2**) # exit **Separate Commands Technique** (config) # map-passall alias **mypass** from 1/2/x1 (config) # map-passall alias **mypass** to **1/2/x2..x5**	
Map Prefix Mode Technique **(config) # map-passall alias gigapass** (config map-passall alias **gigapass**) # from 1/3/x1 (config map-passall alias **gigapass**) # to **mygigastream** (config map-passall alias **gigapass**) # exit **Separate Commands Technique** (config) # map-passall alias **mypass** from 1/3/x1 (config) # map-passall alias **mypass** to **mygigastream**	Configures a map passall from 1/3/x1 to the GigaStream with the alias **mygigastream**.

Related Commands

The following table summarizes other commands related to the **map-passall** command:

Task	Command
Displays all map-passalls.	**# show map-passall**
Displays information for a specified map-passall.	**# show map-passall alias mymap**
Displays all map-passalls.	**# show map-passall all**
Display all map-passalls in table format.	**# show map-passall brief**
Deletes a specified map-passall.	**(config) # no map-passall alias mymap**
Deletes the comments for a specified map-passall.	**(config) # no map-passall alias mymap comment**
Deletes all sources configured for a specified map-	**(config) # no map-passall alias mymap from**

Task	Command
passall.	
Deletes an assigned role from a specified map-passall.	**(config) # no map-passall alias mymap roles assign monitor**
Deletes all assigned roles from a specified map-passall.	**(config) # no map-passall alias mymap roles assign all**
Deletes all destinations configured for a specified map-passall.	**(config) # no map-passall alias mymap to**

map-scollector

Required Command-Line Mode = Configure

Use the **map-scollector** command to configure shared collector maps and parameters. Use a shared collector to send any packets that do not match the map rules to a destination.

The **map-scollector** command has the following syntax:

```
map-scollector alias <alias>
  collector <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias |
      inline-tool-group-alias | inline-serial-alias | bypass>
  comment <comment>
  from <port-id | port-alias | port-list | inline-network-alias | inline-network-group-alias>
rewrite-dstmac <value> | rewrite-<value>
rewrite-dstip <value> | rewrite-srcip <value>
vlan-op add <vlan id> tpid <value>
vlan-op modify <vlan id> tpid <value>
vlan-op strip
  roles <assign | replace> <role> [to <role list>]
```

The following table describes the arguments for the **map-scollector** command:

Argument	Description
alias <alias>	Specifies the name of the shared collector map. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # map-scollector alias scoll**
<port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias | inline-tool-group-	Specifies the destination(s) for packets matching this shared collector map. Use one of the following:

Argument	Description
alias | inline-serial-alias | bypass>	• **port-id, port-alias, port-list**—Sends traffic to one or more tool ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **gigastream-alias, gigastream-alias-list**—Sends traffic to the specified tool GigaStream. Refer to the "*GigaStream*" section in the *GigaVUE Fabric Management Guide* for details on GigaStream. • **inline-tool-alias**—Sends traffic to the specified inline tool alias. • **inline-tool-group-alias**—Sends traffic to the specified inline tool group alias. • **inline-serial-alias**—Sends traffic to the specified inline tool series alias. • **bypass**—Sends traffic to the specified inline bypass. Refer to the *"Associating Inline Networks with Inline Tools Using Inline Maps"* section in the *GigaVUE Fabric Management Guide* for details on **inline-tool-alias**, **inline-tool-group-alias**, **inline-serial-alias**, and **bypass**. For example: **(config) # map-scollector alias scoll 2/1/x1**
comment <comment>	Specifies a unique text string that describes the shared collector map. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks.For example: **(config) # map-scollector alias scoll comment "Shared collector"**
from <port-id | port-alias | port-list | inline-network-alias | inline-network-group-alias>	Specifies the source(s) for packets matching this shared collector map. Use one of the following: • **port-id, port-alias, port-list**—Sends traffic from one or more network ports specified using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **inline-network-alias**—Sends traffic from the specified inline network alias. • **inline-network-group-alias**—Sends traffic from the specified inline network group alias. Refer to the *"Associating Inline Networks with Inline Tools Using Inline Maps"* in the *GigaVUE Fabric Management Guide* for details on **inline-network-alias** and **inline-network-group-alias**. For example: **(config) # map-scollector alias scoll from inNet**
rewrite-dstmac <value> | rewrite-src mac <value>	For MAC Address rewrite ,configure the destination and Source fields as follows:

Argument	Description
	• **rewrite-dstmac xx:xx:xx:xx:xx:xx** — Configure destination MAC rewrite for all the pass rules associated with the map. • **rewrite-srcmac xx:xx:xx:xx:xx:xx**— Configure source MAC rewrite for all the pass rules associated with the map.
rewrite-dstip <value> | rewrite-srcip <value>	For IP Address rewrite, configure the destination and Source fields as follows: • **rewrite-dstip x.x.x.x** — Configure destination IP rewrite for all the pass rules associated with the map. • **rewrite-srcip x.x.x.x**— Configure source IP rewrite for all the pass rules associated with the map.
vlan-op add <vlan id> tpid <value>	For VLAN manipulation, configure a VLAN ID and TPID value: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095. • **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100, and other supported values are 0x9100 and 0x88a8. To unconfigure VLAN manipulation for map-based configuration use the below: • **no vlan-op** — Unconfigure the map level configured VLAN tag.
vlan-op modify <vlan id> tpid <value>	For VLAN manipulation, modify a VLAN ID and TPID value as follows: • **vlan-id** — Specifies a user-defined value for the external VLAN ID in the range of 1 to 4095. • **tpid**— Configures a TPID value for the new VLAN tag. The default value of TPID is 0x8100 and other supported values are 0x9100 and 0x88a8. To unconfigure VLAN manipulation for map-based configuration use the below: • **no vlan-op** — Unconfigure the map level configured VLAN tag.
vlan-op strip	It is used to remove or strip the outer VLAN tag from the incoming packet.
roles <assign | replace> <role> [to <role list>]	Assigns a user role to a map access list or replaces a user role for a shared collector map. For example: **(config) # map-scollector alias scoll roles assign monitor to listen_roles**

Related Commands

The following table summarizes other commands related to the **map-scollector** command:

Task	Command
Displays all shared collector maps.	**# show map-scollector**
Displays information for a specified shared collector map.	**# show map-scollector alias mycoll**
Displays all shared collector maps.	**# show map-scollector all**
Displays all shared collector maps in table format.	**# show map-scollector brief**
Deletes a specified shared collector map.	**(config) # no map-scollector alias mycoll**
Deletes all destinations configured for a specified shared collector map.	**(config) # no map-scollector alias mycoll collector**
Deletes the comments for a specified shared collector map.	**(config) # no map-scollector alias mycoll comment**
Deletes all sources configured for a specified shared collector map.	**(config) # no map-scollector alias mycoll from**
Deletes an assigned role from a specified shared collector map.	**(config) # no map-scollector alias mycoll roles assign monitor**
Deletes all assigned roles from a specified shared collector map.	**(config) # no map-scollector alias mycoll roles assign all**
Deletes the configured destination and source MAC Address.	**(config)# no rewrite-dstmac | no rewrite-srcmac**
Deletes the configured destination and source IP Address.	**(config)# no rewrite-dstip | no rewrite-srcip**
Deletes the VLAN manipulation configured from a specified map.	**(config)# no vlan-op**

map-template

Use the **map-template** command to create map templates and share them with users associated with specified roles. Map-templates provide a way to create generic sets of rules that can be shared with other GigaVUE-OS users. Once a map-template is shared, a user with the rights to use it can create a new map based on the template.

The criteria available for the **map-template** command are the same as those available for the **map** command described inmap. The only differences are that map-templates do not include any network ports, tool ports, or GigaSMART operations. Also, a map-template requires at least one rule.

The **map-template** command has the following syntax:

map-template alias <alias>
comment <comment>

roles <assign | replace> <role> [to <role list>]
rule
add <drop | pass>
delete <all | rule-id <rule ID>>

Refer to map for details.

The **map-template rule** command has the following syntax:

rule add <drop | pass>
bidir
comment <comment>
dscp <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef>
ethertype <2-byte-hex>
inner-ethertype <2-byte-hex>
inner-ipver <4 or 6>
inner-ipfrag <no-frag|all-frag|all-frag-no-first|first-frag|first-or-no-frag>
inner-ipdst <IP address>
inner-ipsrc <IP address>
inner-ip6dst <IPv6 address>
inner-ip6src <IPv6 address>
inner-protocol <1-byte-hex>
inner-portdst <0~65535 | x..y> inner-portdst-subset <even|odd>
inner-portsrc <0~65535 | x..y> inner-portsrc-subset <even|odd>
inner-uda1-data <16-byte-hex> inner-uda1-mask <16-byte-hex> inner-uda1-offset <0-120 bytes>
inner-uda2-data <16-byte-hex> inner-uda2-mask <16-byte-hex> inner-uda2-offset <0-120 bytes>
inner-vlan <vlan | vlan1..vlan2> innervlan-subset <even | odd>
ip6dst <IPv6 address> <IPv6 netmask>
ip6fl <3-byte-hex>
ip6src <IPv6 address> <IPv6 netmask>
ipdst <IP address> <netmask>
ipfrag <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-frag>
ipsrc <IP address> <netmask>
ipver <4 | 6>
macdst <MAC address> <MAC netmask>
macsrc <MAC address> <MAC netmask>
mpls-label-id <0~1048575 | x..y> pos <any | 1~7> mpls-label-id-subset <even|odd>
mpls-label-exp <0~7> pos <any | 1~7>
mpls-label-bos <0/1> pos <any | 1~7>
mpls-label-ttl <0~255 | x..y> pos <any | 1~7> mpls-label-ttl-subset <even|odd>
portdst <0-65535 | x..y> portdst-subset <even | odd>
portsrc <0-65535 | x..y> portsrc-subset <even | odd>
protocol <ipv6-hop | icmp-ipv4 | igmp | ipv4ov4 | tcp | udp | ipv6 | rsvp | gre | icmp-ipv6> <1-byte-hex>
rewrite-dstip <value>
rewrite-dstmac <value>
rewrite-srcip <value>
rewrite-srcmac <value>

tcpctl <1-byte-hex> tcpctlmask <1-byte-hex>
tosval <1-byte-hex>
ttl <ttl | ttl1..ttl2>
uda1-data <16-byte-hex> uda1-mask <16-byte-hex> uda1-offset <2-110 bytes>
uda2-data <16-byte-hex> uda2-mask <16-byte-hex> uda2-offset <2-110 bytes>
vlan <vlan | vlan1..vlan2> vlan-subset <even | odd>

vlan-op add <vlan id> tpid <value>

vlan-op modify <vlan id> tpid <value>

vlan-op strip

Refer to map rule for details.

Related Commands

The following table summarizes other commands related to the **map-template** command:

Task	Command
Deletes a specified map template.	**(config) # no map-template alias mytemplate**
Deletes the comments for a specified map template.	**(config) # no map-template alias mytemplate comment**
Deletes an assigned role from a specified map template.	**(config) # no map-template alias mytemplate roles assign monitor**
Deletes all assigned roles from a specified map template.	**(config) # no map-template alias mytemplate roles assign all**

nhb-profile

Required Command-Line Mode = Admin

Use the **nhb-profile** command to configure a negative heartbeat profile, which is a group of attributes that you can apply to an inline tool to configure the negative heartbeat operation of the inline tool.

This command is only applied to GigaVUE HC Seriesnodes. In a cluster environment, this command is only applied to GigaVUE HC Series nodes through the cluster leader.

The content of a negative heartbeat is configurable using the same PCAP file mechanism as for a custom heartbeat packet in a heartbeat profile. Refer to hb-profile.

Also refer to inline-tool for information on enabling negative heartbeat and associating a negative heartbeat profile with an inline tool.

The maximum number of negative heartbeat profiles supported is equal to the maximum number of inline tools, which is 48 on the GigaVUE-HC3, and 8 on the GigaVUE-HC1.

This command is used in the inline bypass solutions described in Configure Inline Bypass Solutions and in the flexible inline arrangements described in Configure Flexible Inline Arrangements.

The **nhb-profile** command has the following syntax:

```
nhb-profile alias <alias>
  custom-packet <URL of PCAP file | none>
  direction <a-to-b | b-to-a | bi-directional>
  period <period>
  recovery-time <recovery time>
```

The following table describes the arguments for the **nhb-profile** command.

Argument	Description
alias <alias>	Specifies the name of the negative heartbeat profile. Use the alias to configure a negative heartbeat profile to associate with an inline tool. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. Some of the parameters for **nhb-profile** have default values, so you can configure a negative heartbeat profile by providing an alias for it, as well as a PCAP file. For example: **(config) # nhb-profile alias nhb_1** **(config nhb-profile alias nhb_1) # custom-packet http://remote/home/nhb.pcap** **(config nhb-profile alias nhb_1) # exit** **(config) #**
custom-packet <URL of PCAP file | none>	Specifies the URL of the custom heartbeat packet, downloaded from a PCAP file, or **none**. The default is **none**. **NOTE:** Unlike **hb-profile**, there is no standard ICMP ARP packet, so a custom packet must always be specified through a PCAP file. The size of a custom heartbeat packet must be less than 128 bytes. If the PCAP file contains several packets, the first packet present in the file is taken as the negative heartbeat packet. For example: **(config nhb-profile alias nhb_1) # custom-packet http://remote/home/nhb.pcap** The PCAP file must be valid before the negative heartbeat profile can be associated with an inline tool. If you are specifying a negative heartbeat packet as well as a custom heartbeat packet, do not use the same PCAP file for both. The supported formats for download are HTTP, HTTPS, FTP, TFTP, SCP, and SFTP. Use the **show nhb-profile** command to display the name of the PCAP file from which the custom heartbeat packet was imported.
direction <a-to-b | b-to-a |	Specifies the direction of the negative heartbeat packet as follows: • **a-to-b**—Specifies from side A to side B of the inline tool.

Argument	Description
bi-directional>	• **b-to-a**—Specifies from side B to side A of the inline tool. • **bi-directional**—Specifies both directions. The default is **bi-directional**. For example: **(config nhb-profile alias nhb_1) # direction a-to-b**
period <period>	Specifies the period of the negative heartbeat packet. This is the number of milliseconds between sending subsequent negative heartbeat packets. The range is from 30 to 5000 milliseconds. The default is 1000 milliseconds. For example: **(config nhb-profile alias nhb_1) # period 600**
recovery-time <recovery time>	Specifies the recovery time of the negative heartbeat packet. This is the minimum number of seconds, since the last negative heartbeat packet was received, to declare that the inline tool is up. The range is from 5 to 60 seconds. The default is 30 seconds. The recovery time starts when the negative heartbeat is first enabled on the inline tool and then every time a negative heartbeat is received. For example: **(config nhb-profile alias nhb_1) # recovery-time 60**

Related Commands

The following table summarizes other commands related to the **nhb-profile** command:

Task	Command
Displays all negative heartbeat profiles.	**# show nhb-profile**
Displays a specified negative heartbeat profile.	**# show nhb-profile alias nhb_1**
Displays all negative heartbeat profiles.	**# show nhb-profile all**
Deletes a specified negative heartbeat profile.	**(config) # no nhb-profile alias nhb_1**
Deletes a custom packet associated with the negative heartbeat profile.	**(config) # no nhb-profile alias nhb_1 custom-packet**
Deletes all negative heartbeat profiles.	**(config) # no nhb-profile all**

no

Required Command-Line Mode = Configure

Use the **no** command to clear, delete, or reset configuration settings in the GigaVUE-OS. Do this by prefacing the corresponding configuration command with the word **no**. For example, **no map alias mymap** deletes the map named **mymap**.

NOTE: The GigaVUE HC Series replaces the **delete** command with the **no** command common to Cisco products.

The **no** prefix is available for the following configuration commands. Refer to the corresponding configuration commands for usage.

aaa	hb-profile	map-scollector	ssh
apps	hostname	map-template	stack-link
banner	ib-pathway	nhb-profile	system
bond	image	notifications	system-health
boot	inline-network	ntp	tacacs-server
card (GigaVUE-OS® HC Series)	inline-network-group	pcap	terminal
chassis	inline-serial	policy	tool-mirror
cli	inline-tool	port	traffic (refer to no traffic)
clock	inline-tool-group	port-group	tunnel
cluster	interface	port-pair	tunnel-endpoint
configure	ip	ptp	ip interface
crypto	ipv6	radius-server	username
email	job	redundancy-profile	vport
filter-template	ldap	serial	web
gigasmart	logging	service (refer to no service)	web
gigastream	map	sfp	
gsgroup	map-group	snmp-server	
gsop	map-passall	spine-link	

no service

Required Command-Line Mode = Configure

Use the **no service** command to disable TCP or UDP small server services.

NOTE: There is not a corresponding configuration command.

The **no service** command has the following syntax:

no service <tcp-small-servers | udp-small-servers>

The following table describes the arguments for the **no service** command:

Argument	Description
no service tcp-small-servers	Disables all small server services on TCP (including echo, chargen, discard, daytime, and time). For example: **(config) # no service tcp-small-servers**

Argument	Description
no service udp-small-servers	Disables all small server services on UDP (including echo, chargen, discard, daytime, and time). For example: **(config) # no service udp-small-servers**

no traffic

Required Command-Line Mode = Configure

Use the **no traffic** command to delete traffic configuration.

> **NOTE:** There is not a corresponding configuration command.

The **no traffic** command has the following syntax:

```
no traffic
  all [keep-stack]
```

The following table describes the arguments for the **no traffic** command:

Argument	Description
no traffic all	Deletes all traffic configuration and resets the port types. For example: **(config) # no traffic all**
no traffic all [keep-stack]	Deletes all traffic configuration, resets the port types, but keeps the stack configuration, including stack ports and GigaStream. For example: **(config) # no traffic all keep-stack** Use this command with Inband Cluster Management so the control messages, exchanged through stack ports and GigaStream, is retained.

notifications

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **notifications** command to configure notification settings. The notification receiver target is the consumer of the notifications, such as GigaVUE-FM. Log in to each GigaVUE-OS node individually to configure the settings.

The **notifications** command has the following syntax:

```
notifications
  enable
  target host <IPv4 address or hostname> port <port ID> <secure | non-secure> username
<username>
      password <password>>
```

The following table describes the arguments for the **notifications** command:

Command	Description
enable	Enables notifications on a particular GigaVUE-OS node. The default is enabled. For example: **(config) # notifications enable**
target host <IPv4 address or hostname> port <port ID> <secure | non-secure> username <username> password <password>>	Configures the notification receiver target as follows: • **host**—Specifies the IPv4 address or the hostname of the notification receiver target. Up to three IP addresses or hostnames, or a combination of both IP addresses and hostnames can be specified, but only one per CLI invocation. • **port**—Specifies the port ID. The range is from 1 to 65535. The default is 5672. **NOTE:** Open port 5672 on GigaVUE-FM so it is accessible for devices to connect to. • **secure | non-secure**—Specifies a secure (TLS-protected) or non-secured (open) connection. • **username**—Specifies the username of the notification receiver. • **password**—Specifies the password of the notification receiver. In the following example, the user will be prompted for the password: **(config) # notifications target ip 1.1.1.1 port 222 secure username user1** In the following example, the default port 5672 will be used and the user will be prompted for the password: **(config) # notifications target ip 1.1.1.1 non-secure username user1** In the following example, the user enters the password: **(config) # notifications target ip 1.1.1.1 port 5672 secure username user1 password PW1**

Related Commands

The following table summarizes other commands related to the **notifications** command:

Task	Command
Displays notification settings and connection status.	**# show notifications**
Disables notifications on a particular GigaVUE-OS node.	**(config) # no notifications enable**
Deletes a specified notification receiver target.	**(config) # no notifications enable target ip 1.1.1.1 port 5672**

ntp

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **ntp** command to configure persistent synchronization of the GigaVUE HC Series node's system clock with an NTP server. Separate arguments let you add servers and enable the use of NTP generally. You can also use the **ntpdate** command to perform one-time synchronization of the system clock. Refer to ntpdate for more information.

The **ntp** command has the following syntax:

```
ntp
  authentication enable
  authentication-key <key number>
  disable
  enable
  server <hostname, IPv4 or IPv6 address> [disable | key <key number> | keys enable | version
<version
      number>]
```

The following table describes the arguments for the **ntp** command:

Argument	Description
authentication enable	Enables NTP authentication. For example: **(config) # ntp authentication enable**
authentication-key <key number>	Specifies an NTP authentication key.
disable	Disables the use of NTP for synchronization of the system's clock. For example: **(config) # ntp disable**
enable	Enables the use of NTP for synchronization

Argument	Description
	of the system's clock. For example: **(config) # ntp enable**
server <hostname, IPv4 or IPv6 address> [disable | key <key number> | keys enable | version <version number>]	Adds an NTP server to the GigaVUE HC Seriesnode's list. The arguments are as follows: • **hostname**—Specifies a DNS name of the server, such as: time.windows.com • **IPv4 or IPv6**—Specifies the IP address of the server. • **disable**—Temporarily disables the specified server. • **key**—Specifies the NTP key number of this server • **keys**—Enables the NTP key for this server • **version**—Specifies the NTP version in use by the specified server (3 or 4). For example, the following command adds an NTP server on an IPv4 address: **(config) # ntp server 192.168.1.10**

Related Commands

The following table summarizes other commands related to the **ntp** command:

Task	Command
Displays detailed information on current NTP settings.	**# show ntp**
Displays NTP configuration.	**# show ntp configured**
Disables NTP authentication.	**(config) # no ntp authentication enable**
Deletes a specified NTP authentication key.	**(config) # no ntp authentication-key <key number>**
Deletes a specified trusted NTP authentication key.	**(config) # no ntp authentication-key <key number> trusted**
Enables NTP.	**(config) # no ntp disable**
Disables NTP.	**(config) # no ntp enable**
Deletes the specified NTP server by IPv4 address.	**(config) # no ntp server 1.1.1.1**

Task	Command
Deletes the specified NTP server by hostname.	**(config) # no ntp server time.windows.com**
Re-enables a specified NTP server.	**(config) # no ntp server 1.1.1.1 disable**
Disables the key for the specified NTP server.	**(config) # no ntp server 1.1.1.1 keys enable**

ntpdate

Required Command-Line Mode = Enable

Required User Level = Admin

Use the ntpdate command to perform one-time synchronization of the GigaVUE HC Series node's system clock with a specified NTP server. This contrasts with the **ntp server** command which configures persistent NTP synchronization.

The **ntpdate** command has the following syntax:

ntpdate <hostname, IPv4 or IPv6 address>

For example:

(config) # ntpdate time.nist.gov

onie

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **onie** command to reboot a Certified Traffic Aggregation White Box (white box) into Open Network Install Environment (ONIE) modes of debug, reinstall, uninstall, and update. ONIE automatically discovers, fetches, and executes the GigaVUE-OS installer.

This command is only available on white boxes on which GigaVUE-OS is installed.

The **onie** command has the following syntax:

onie reboot mode <debug | reinstall | uninstall | update>

The following table describes the arguments for the **onie** command:

Argument	Description
reboot mode <debug |	Specifies the ONIE reboot modes as follows:

Argument	Description
reinstall | uninstall | update>	• **debug**—reboots the white box into ONIE debug mode. • **reinstall**—reboots the white box into ONIE reinstall mode. • **uninstall**—reboots the white box into ONIE uninstall mode. Uninstall wipes off all non-ONIE reserved regions of NOR flash and storage block device. • **update**—reboots the white box into ONIE update mode used for updating the ONIE firmware. For example: **(config) # onie reboot mode debug** **(config) # onie reboot mode reinstall** **(config) # onie reboot mode uninstall** **(config) # onie reboot mode update**

pcap

Required Command-Line Mode = Admin

Use the **pcap** command to configure packet capture, which lets you capture packets at an ingress port, an egress port, or both and the captured packets are stored in a PCAP file.

To configure packet capture, define filters to capture specific traffic based on rules. The following criteria can be specified in the rules:

- Source MAC address
- Destination MAC address
- VLAN ID
- Inner-VLAN
- Layer 2 ethernet type
- Source IPv4 address
- Destination IPv4 address
- Internet protocol
- IP version number
- IP fragmentation bits
- Time to Live (TTL) value
- DiffServ Code Point bits
- Layer 4 destination port number
- Layer 4 source port number
- TCP flags

Packet capture is supported on GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3, and GigaVUE TA Series nodes. It is supported on both standalone nodes and clusters.

The port type used for packet capture can be tool, network, hybrid, inline tool, or inline network. They must be physical ports.

Refer to the following notes:

- The PCAP feature supports up to 16 capture files per device. Each capture file can have up to 40000 packets. A capture file is maintained per PCAP session. Each session can have up to 64 filter rules per direction. Each capture file can be viewed, deleted, or uploaded out of the device for offline use.
- The PCAP feature supports up to 16 active capture sessions at a time per port on all GigaVUE platforms except GigaVUE-TA400. The GigaVUE-TA400 platform currently supports one active PCAP session per port at a time.
- The PCAP configuration doesn't persist across node reboots or upgrades.
- The PCAP feature is not supported on stack ports in legacy stacking mode.
- The PCAP feature is not supported on ports associated with the IP interface.
- The PCAP feature does not support 'vlan' and 'inner-vlan' filter rules on a tool or hybrid port in the 'tx' direction.
- The PCAP feature on tool ports does not capture the vlan tag specified with the ingress-vlan-tag feature. To overcome this, redirect the traffic to another hybrid port along with other tool ports and capture the packets on the hybrid port ingress.
- The PCAP feature on GigaVUE-HC1-Plus, GigaVUE-HC1, GigaVUE-TA25, GigaVUE-TA25E, and GigaVUE-TA400 platforms contain extra VLAN header added in capture files. Untagged packet captures contain vlan-tag 1 header added and tagged packet captures contain an outer tag duplicated.
- The PCAP feature on port discovery protocols (LLDP/CDP/GDP) enabled ports will not capture the discovery protocol control packet in the PCAP file.
- The PCAP feature may miss some packets in the capture file depending on the rate of traffic being captured.

The **pcap** command has the following syntax:

```
pcap   alias <alias>
    channel-port <port ID>
    packet-limit <1-20000>
    port <port ID> <tx | rx | both>
    filter
      dscp <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef>
      ethertype <2-byte-hex>
```

inner-vlan <vlan>
ipdst <IP address> <netmask>
ipfrag <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-frag>
ipsrc <IP address> <netmask>
ipver <4 | 6>
macdst <MAC address> <MAC netmask>
macsrc <MAC address> <MAC netmask>
portdst <0-65535>
portsrc <0-65535>
protocol <ipv6-hop | icmp-ipv4 | igmp | ipv4ov4 | tcp | udp | ipv6 | rsvp | gre | icmp-ipv6>
tcpctl <1-byte-hex>
ttl <ttl>
vlan <vlan>

The following table describes the arguments for the **pcap** command:

Argument	Description
alias <alias>	Specifies the name of the packet capture filter. For example: **(config) # pcap alias issl_ack**
channel-port <port ID>	Specifies the channel port identifier for the packet capture filter, in the format <bid/sid/pid>. The channel port can be a network, tool, or hybrid port. The channel port is any unused port. Unused means that it does not have any map configuration. In addition, the channel port must be on the same node as the capture port. Finally, the channel port must be administratively enabled and must remain enabled while a packet capture filter is configured. You must specify one channel port for each **tx** or **both** direction. A channel port is not needed for **rx**. For example: **(config pcap alias issl_ack) # channel-port 1/1/x2** **(config) # port 1/1/x2 params admin enable** **NOTE:** If a PCAP configuration is deleted, the channel ports configured in the PCAP will go down.
packet-limit <1-40000>	Specifies the number of packets to capture. The valid range is 1 to 40000 for all the platforms. Use the packet limit to specify that the packet capture will stop after the specified number of packets have been captured. The default value is 40000 for all the platforms. For example: **(config pcap alias issl_ack) # packet-limit 100** If you do not specify a packet limit, delete the packet capture filter to stop capturing. For example:

Argument	Description
	(config) # no pcap alias issl_ack
port <port ID> <tx | rx | both>	Specifies the port identifier for the packet capture filter, in the format <bid/sid/pid>, and the direction as follows: • **tx**—Specifies the transmitting end (egress). • **rx**—Specifies the receiving end (ingress). • **both**—Specifies both the transmitting and the receiving ends (egress and ingress). This port may also be referred to as the capture port or the filter port. The port type can be tool, network, hybrid, inline tool, or inline network. They must be physical ports. Examples: **(config pcap alias issl_ack) # port 1/1/x1 tx**
filter **dscp <af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef>** **ethertype <2-byte-hex>** **inner-vlan <vlan>** **ipdst <IP address> <netmask>** **ipfrag <no-frag | all-frag | all-frag-no-first | first-frag | first-or-no-frag>** **ipsrc <IP address> <netmask>** **ipver <4 | 6>** **macdst <MAC address> <MAC netmask>** **macsrc <MAC address> <MAC netmask>** **portdst <0-65535>** **portsrc <0-65535>** **protocol <ipv6-hop | icmp-ipv4 | igmp | ipv4ov4 | tcp | udp | ipv6 | rsvp |**	Specifies the rules on which to filter traffic as follows: • **dscp**—Specifies the decimal DSCP value. You can select any value within the four Assured Forwarding (af) class ranges or ef for Expedited Forwarding (the highest priority in the DSCP model). The valid DSCP values by Assured Forwarding Class are as follows: o Class 1—11, 12, 13 o Class 2—21, 22, 23 o Class 3—31, 32, 33 o Class 4—41, 42, 43 o Expedited Forwarding—ef • **ethertype**—Specifies the layer 2 ethernet type value. • **inner-vlan**—Specifies the VLAN ID value as a number between 1 and 4094. • **ipdst**—Specifies the destination IPv4 address and IP mask or a wildcard with an IP mask. • **ipfrag**—Specifies any of the IP fragments listed below. o no-frag—Matches unfragmented packets. o all-frag—Matches any fragment. o all-frag-no-first—Matches all fragments except the first fragment in a packet. o first-frag—Matches the first fragment of a packet. o first-or-no-frag—Matches unfragmented packets or the first fragment of a packet • **ipsrc**—Specifies the source IPv4 address and IP mask or a wildcard with an IP mask. • **ipver**—Specifies the IP version for traffic, either IPv4 or IPv6. • **macdst**—Specifies the destination MAC address and MAC netmask. • **macsrc**—Specifies the source MAC address and MAC netmask. • **portdst**—Specifies the Layer 4 destination port number, from 0 to 65535. A range of ports is not supported.

Argument	Description
gre | icmp-ipv6> **tcpctl <1-byte-hex>** **ttl <ttl>** **vlan <vlan>**	• **portsrc**—Specifies the Layer 4 source port number, from 0 to 65535. A range of ports is not supported. • **protocol**—Specifies the Internet protocol. The valid protocols and their hex value are as follows: o ipv6-hop (0x0) o icmp-ipv4 (0x1) o igmp (0x2) o ipv4ov4 (0x4) o tcp (0x6) o udp (0x11) o ipv6 (0x29) o rsvp (0x2E) o gre (0x2F) o icmp-ipv6 (0x3A) o A custom-defined value can also be defined in 1 byte hex. • **tcpctl**—Specifies TCP control bits, such as SYN, FIN, ACK, URG, as 1 byte hex values. Rules using the **tcpctl** parameter must also specify the protocol as **tcp**. • **ttl**—Specifies the Time to Live (TTL—IPv4) or Hop Limit (IPv6) value in an IP packet, as a number between 0 and 255. • **vlan**—Specifies the VLAN ID value as a number between 1 and 4094. You can configure multiple filter rules to the same PCAP. For example: **(config pcap alias issl_ack) # filter ipsrc 10.10.1.16 /24 portsrc 2152 protocol udp**

Related Commands

The following table summarizes other commands related to the **pcap** command:

Task	Command
Displays all packet capture filters.	**# show pcap**
Displays a specified packet capture filter.	**# show pcap alias issl_ack**
Displays PCAP files.	**show files pcap**
Sends a PCAP file to a remote host. Refer to file.	**(config) # file pcap upload pcap_p1_2018_05_08_17_28.pcap scp://myNode@10.115.0.100/tftpboot/myName/.**
Stops a specified packet capture and deletes it.	**(config) # no pcap alias issl_ack**

ping

Required Command-Line Mode = Enable

Use the **ping** command to send a standard ICMP ping message from the **Mgmt** port. You can ping both IPv4 and IPv6 systems.

The **ping** command has the following syntax:

```
ping [-LRUbdfnqrvVaA] [-c count] [-i interval] [-w deadline]
     [-p pattern] [-s packetsize] [-t ttl] [-I interface or address]
     [-M mtu discovery hint] [-S sndbuf]
     [ -T timestamp option ] [ -Q tos ] [hop1 ...] destination
```

These are standard Linux options for **ping**. Refer to online man pages for details.

ping6

Required Command-Line Mode = Enable

Use the **ping6** command to send a standard ICMPv6 ping message from the **Mgmt** port.

The **ping6** command has the following syntax:

```
ping [-LUdfnqrvVaA] [-c count] [-i interval] [-w deadline]
     [-p pattern] [-s packetsize] [-t ttl] [-I interface]
     [-M mtu discovery hint] [-S sndbuf]
     [-F flow label] [-Q traffic class] [hop1 ...] destination
```

These are standard Linux options for **ping6**. Refer to online man pages for details.

pld

Required Command-Line Mode = Configure

Use the **pld** command to upgrade programmable logic devices (PLDs) such as field programmable gate arrays (FPGAs) in GigaVUE-HC3, G-TAP A Series 2, GigaVUE-HCT, and GigaVUE-HC1-Plus.

The following components in GigaVUE-HC3 have an FPGA:

- PRT-HC3-X24 module
- PRT-HC3-C08Q08 module
- PRT-HC3-C16 module
- SMT-HC3-C05 module
- BPS-HC3-C25F2G module
- BPS-HC3-Q35C2G module
- BPS-HC3-C35C2G module

- SMT-HC3-C08 module
- Control card (also referred to as the main board)

The following components in GigaVUE-HC1-Plus have an FPGA:

- HC1P-C04X08 module
- BPS-HC1-D25A60 module
- BPS-HC1-D35C60 module
- SMT-HC1A- R module
- Control card

The following components in GigaVUE-HCT have an FPGA:

- BPS-HC1-D25A60 module
- BPS-HC1-D35C60 module
- Control/CPU card
- PRT-HC1-G12

FPGA images are bundled with the software image and upgraded with the software image upgrade; however, the **pld** command allows you to upgrade each FPGA individually. Refer to theimage for information on downloading the software image using the **image fetch** command. After a PLD upgrade, the node must be hard reloaded. This is also known as a hard power recycle. Issue the **show pld** command to display the PLDs that need to be upgraded, then only upgrade those.

The **pld** command for GigaVUE-HC3, GigaVUE-HC1-Plus, and GigaVUE-HCT has the following syntax:

```
pld
  upgrade slot <slot ID>
```

The **pld** command for G-TAP A Series 2 has the following syntax:

```
pld
  upgrade
```

The following table describes the arguments for the **pld** command :

Argument	Description
• GigaVUE-HC3, GigaVUE-HC1-Plus, and GigaVUE-HCT **upgrade slot <slot ID>**	Upgrades the PLD image in the specified slot. For example: **(config) # pld upgrade slot 2** To upgrade the PLD image on the control card: **(config) # pld upgrade slot cc1**
• G-TAP A Series 2 **(config) # pld upgrade**	Upgrades the PLD image

Related Commands

The following table summarizes other commands related to the **pld** command:

Task	Command
Displays information about PLDs, such as, the current revision, and if an upgrade is needed or not.	**show pld**
Displays information about PLDs for a specific slot ID in GigaVUE-HC3.	**show pld slot 2**

proxy-server profile

Required Command-Line Mode = Admin

Use the **proxy-server profile** command to set up a proxy server profile, which enables the routing of specific internet traffic through a proxy. This includes URL category downloads, lookups, CRL revocations, and OCSP requests/responses.

This command is supported in SMT-HC3-C08 module on GigaVUE-HC3,SMT-HC1-S module on GigaVUE-HC1 and GigaVUE-HC1-Plus.

NOTE:
1. Only IPv4 addresses are supported.
2. A maximum of ten proxy-server profiles can be configured.
3. When setting up the proxy-server profile, make sure to use a unique profile alias for each unique proxy-server configuration (proxy address, port, etc.) across all devices managed by GigaVUE-FM.
4. If multiple devices require the same proxy-server profile settings, use the same profile alias for those devices.
5. If you upgraded your device from GigaVUE-OS version 6.8.xx to 6.9.xx, the Proxy Server profiles configured through GigaVUE-OS CLI will be migrated to GigaVUE-FM. However, you will need to reconfigure your password in GigaVUE-FM if you intend to use this proxy server profile in a new solution. Any existing solutions will be migrated successfully along with the password.

The **proxy-server profile** command has the following syntax:

proxy-server profile alias <profile1>
protocol http
proxy-address <ipv4-address | domain name>
port <l4-port>
auth-type [none | basic]
auth-username <user-name>
auth-password [<cr> | <clear-text password>] # when hit enter, the password will be obtained as masked.
periodic-ping [enable | disable] #default: disable
periodic-ping type http-connect
periodic-ping interval <1 - 10; default 5> #duration in minutes periodic-ping failure-retry <1 - 3; default 3>
exit

The following table describes the arguments for the **proxy-server profile** command.

Argument	Description
proxy-server profile alias <profile1>	Specifies the name of the proxy-server profile.
protocol http	Specifies the protocol **http**. The http protocol supported is only access type.
proxy-address <ipv4-address | domain name>	Specifies the proxy-address as follows: • **ipv4-address**—Specifies the IPv4 address. • **domain name**—Specifies the domain name.
port <port>	Specifies the port.
auth-type [none | basic]	Specifies the authorization type.

Argument	Description
auth-username <user-name>	Specifies the authorization profile name.
auth-password [<cr> | <clear-text password>]	Specifies the authorization profile password.
periodic-ping [enable | disable]	Specifies enabling the periodic ping. By default this will be disabled by default.
periodic-ping type http-connect	Specifies the Periodic ping type.
periodic-ping interval <1 - 10; default 5>	Specifies the time interval at which the periodic ping should be sent. The time interval is from one to ten minutes. By default, five minutes will be configure.
periodic-ping failure-retry <1 - 3; default 3>	Specifies the number of times the periodic ping will be re-attempted. The default is three.

Related Commands

The following table summarizes other commands related to the **proxy-server profile** command:

Task	Command
Displays the proxy-server profiles.	**# show proxy-server profile [all | alias <alias-name>]**
Displays ARP information for the GigaSMART engine port.	**# show gigasmart engine 1/1/e1 arp**

policy

Use the **policy** command to tie actions and conditions together into an Active Visibility policy. Active visibility is a framework designed to react to events and take actions in response. When conditions change, actions are triggered as specified by policies. Refer to the *"Configuring Active Visibility"* section in the *GigaVUE Fabric Management Guide* for details.

When a policy is triggered, an SNMP event can optionally be generated.

The **policy** command has the following syntax:

```
policy
   alias <alias>
     action
        add <action name> [param <param name> <param value>] .. [param <param name>
<param value>]
        delete <action ID>
     comment <comment>
```

```
condition
  add <condition name> [param <param name> <param value>] .. [param <param name>
     <param value>]
  delete <condition ID>
enable
reset
all <enable |  reset>
```

The following table describes the arguments for the **policy** command:

Argument	Description
alias <alias>	Specifies the name of the policy. Up to 100 polices per cluster can be created. For example: **(config) # policy alias p1**
action **add <action name> [param <param name> <param value>] .. [param <param name> <param value>]**	Adds an action to the policy. The actions are predefined. Up to five (5) actions can be specified in a policy. The parameter names and values that need to be specified, depend on the action. The following are the keywords that are used as actions: • **PortEnable** • **MapDisable** • **PortFilterAdd** • **InlineSslFlushAll**- Clearing of session table entries. • **InlineSslFlush:**- Clearing of specific gsGroup session entries alone. The following are the keywords that are used as parameters in some of the actions, as well as the definition of the value: • **gsGroup** - Alias of the GigaSMART group. Refer gsgroup to create GigaSMART group. • **mapAlias** - The map alias. Map aliases specified in actions are not validated. Ensure that they exist and are valid. For details, refer to Specifying Keyword, mapAlias. • **policyAlias** - The policy alias. Policy aliases specified in actions are not validated. Ensure that they exist and are valid. For details, refer to Specifying Keyword, policyAlias. • **portId** - The port identifier, in the one of the following formats: o single port—a/b/c o multiple ports, separated by commas—a1/b1/c1,a2/b2/c2 o range of ports—a/b/c..d o Port identifiers specified in actions are not validated. Ensure that they exist and are valid.

Argument	Description
	o Port aliases and GigaStream aliases are not supported. Also, the argument "any" is not a supported argument for actions. For details, refer to Specifying Keyword, portId. • **ruleId -** The rule identifier, from a map. For details, refer to Specifying Keyword, ruleId. • **ruleStr** - The map rule string. For details, refer to Specifying Keyword, ruleStr. • **inlineNetAlias** - The inline network port alias. For details, refer to Specifying Keyword, inlineNetAlias. • **inlineToolAlias** - The inline tool port alias. For details, refer to Specifying Keyword, inlineToolAlias. • **inlineNetTrafficPath** - The traffic path for the inline network. The values are: o **to-inline-tool** - Traffic is forwarded to the inline tool. o **bypass** - Traffic bypasses the inline tool. o **drop** - Traffic is dropped at the inline tool. o **monitoring** - Traffic is fed to the inline tool and absorbed, while a copy of the traffic is sent to the next inline tool in the sequence. Traffic returned from side B of the network is also absorbed at the inline tool in monitoring mode. For details, refer to Specifying Keyword, inlineNetTrafficPath. • **oobFromAlias** - The out-of-band inline network alias specified when you add or remove out-of-band copy for flexinline map. For details, refer to Specifying Keyword, oobFromAlias. • **oobDir** - The direction of traffic specified when you add or remove out-of-band copy for flexinline map. The values are: o **a-to-b**—Taps traffic from the a-to-b side of the source. o **b-to-a**—Taps traffic from the b-to-a side of the source. For details, refer to Specifying Keyword, oobDir. • **oobTag** - The tag specified for out-of-band tool port. The values are: o **none**—Does not tag packets that are going to the OOB tool. The default is none. o **as-inline**—Uses the same VLAN tag that was configured for the flexible inline map. o **original**—Uses the original VLAN tag of the packet received from the inline network. For details, refer to Specifying Keyword, oobTag. A policy is triggered if all the conditions are met, then all the actions are executed. Examples: **(config) # policy alias p1 action add PortEnable param portId 1/1/x1** **(config) # policy alias p1 action add MapDisable param mapId m1**

Argument	Description
	(config) # policy alias AnyPortUp action add PortFilterAdd param portId &PortUp.portId& param ruleStr "pass vlan 100"
delete <action ID>	Deletes an action from the policy. The policy must exist. To find out the action ID, type ? as follows: **(config) # (config) # policy alias p1 action delete ?** The existing actions will be listed. For example: **(config) # (config) # policy alias p1 action delete 2**
comment <comment>	Specifies a unique text string that describes the policy. Comments can be up to 256 characters. Comments must be enclosed in double quotation marks. Comments can be added only after a policy has been created. For example: **(config) # policy alias p1 comment "Thursday policy"**
condition add	Adds a condition to the policy. The conditions are predefined. Up to five (5) conditions can be specified in a policy. The parameter names and values that need to be specified, depend on the condition. The following are the keywords that are used as conditions: • **PortTxPktLow -** Packets transmitted from the port is minimum. • **PortTxPktHigh -** Packets transmitted from the port is maximum • **PortRxPktLow** - Packets received at the port is minimum. • **PortTxPktHigh**- Packets received at the port is maximum. The template specifies the parameters and values that must be included in the definition of a condition. The following are the keywords that are used as parameters in some of the conditions, as well as the definition of the value: • **gsGroup** - Alias of the GigaSMART group. Refer gsgroup to create GigaSMART group. • **inlineToolAlias** - Alias of the inline tool. For CLI commands to create an inline tool, refer to inline-tool • **inlineToolGrpAlias** - Alias of the inline tool group. For CLI commands to create an inline tool group, refer to inline-tool-group • **period** - The number of seconds from 1 to 7200 (integers only). For details, refer to Specifying Keyword, period on page 153. • **thresh** - The threshold value from 0 to the maximum (a 64-bit number). For details, refer to Specifying Keyword, thresh on page 154. • **threshPct** - The percentage threshold value from 0 to 100. For details, refer to Specifying Keyword, threshPct on page 154. • **timeStr -** The time string, specified in the Cron format "(a b c d e f)".

Argument	Description
	For details, refer to Specifying Keyword, timeStr on page 154. • **portId** - The port identifier, in the one of the following formats: ▪ single port—a/b/c ▪ multiple ports, separated by commas—a1/b1/c1,a2/b2/c2 ▪ range of ports—a/b/c..d ▪ any port—any(a/b/c..d), which includes the keyword, any ▪ Port identifiers specified in conditions are not validated. Ensure that they exist and are valid. ▪ Port aliases and GigaStream aliases are not supported. For details, refer to Specifying Keyword, portId on page 155. The policy is executed only when all conditions are met. There is only one unique condition per policy. Examples: **(config) # policy alias OverloadedToolPort condition add PortTxUtilHigh param portId 1/1/x1 param threshPct 80** **(config) # policy alias AnyPortUp condition add PortUp param portId any(3/1/q4..q6) param period 5** **(config) # policy alias SaveMemory condition add TimeOfDay param timeStr "(45 10 * * * *)"**
delete <condition ID>	Deletes a condition from the policy. The policy must exist. To find out the condition ID, type ? as follows: **(config) # (config) # policy alias p1 condition delete ?** The existing conditions will be listed. For example: **(config) # (config) # policy alias p1 condition delete 10**
enable	Enables the Active Visibility policy. To be executed, a policy must be enabled. For example: **(config) # policy alias p1 enable**
reset	Resets the status of the Active Visibility policy. For example: **(config) # policy alias p1 reset**
all <enable | reset>	Enables all policies or resets all policies. Examples: **(config) # policy all enable** **(config) # policy all reset**

Related Commands

The following table summarizes other commands related to the **policy** command:

Task	Command
Displays all actions in brief format.	**# show action**
Displays a specified action in detail.	**# show action alias MapRuleAdd**
Displays all actions in detail.	**# show action detail**
Displays all conditions in brief format.	**# show condition**
Displays a specified condition in detail.	**# show condition alias PortRxUtilHigh**
Displays all conditions in detail. In the output of the **show condition** command: • The conditions from PortDown through PortUp are port-based conditions. Use them to monitor link state, port utilization, or packet counts (discards, drops, or errors). • The conditions from TimeFriday through TimeWeekend are time-based conditions. Use them for scheduling. • The Template column displays the parameters that must be specified when defining the condition in a policy. The template contains keywords, some which are mandatory, and others which are optional. For example, the keyword <$portId$> is mandatory, while [$period$] is optional. The strings enclosed in dollar signs ($) are parameters that will be specified as part of configuring a policy. Some conditions do not have any parameters, such as TimeWeekday. When the node is up, the PortUp and PortDown conditions are evaluated immediately upon a link status change while the port TX and RX conditions are evaluated every five seconds through polling. When the node reboots or switches over, refer to Polling Following a Node Reboot, Switchover, or Cluster Change.	**# show condition detail**
Displays all specified Active Visibility policies. The output of the show policy command displays the following: • if a policy has been triggered • how many times a policy has run • the last time a policy has run, which provides a history of the last 5 executions	**# show policy**
Displays a specified policy	**# show policy alias policy1**
Displays all policies in detailed format. **NOTE:** In the output of the **show policy detail** command, the	**# show policy detail**

Task	Command
following is displayed if the Active Visibility Daemon has initialized, but is not yet polling: **Active Visibility Daemon is not polling!! Please try again in 222 seconds.** The message indicates when the Active Visibility Daemon will start polling following a node or cluster reboot or a cluster leader change. If polling is not ready, the output of the **show policy detail** command displays the following message: **Active Visibility Daemon is not polling!! Please try again in 133 seconds.** Following a node reboot, switchover, or a change in the cluster, such as a role change, there is a period of time before the polling mechanism starts. The period of time depends on certain variables such as the number of ports on the node and the size of the cluster in which the node is a part of. If there is a large number of ports or if the node is part of a large cluster, the polling mechanism may take a longer period of time to start. In general, the period of time is between 10 and 15 minutes.	
Deletes a specified policy.	**(config) # no policy alias policy1**
Deletes a comment for a specified policy.	**(config) # no policy alias policy1 comment**
Deletes a comment for a specified policy.	**(config) # policy alias policy1 no comment**
Disables the specified policy.	**(config) # no policy alias policy1 enable**
Disables the specified policy.	**(config) # policy alias policy1 no enable**
Deletes all policies.	**(config) # no policy all**
Disables all policies.	**(config) # no policy all enable**

port

Required Command-Line Mode = Configure

Use the **port** command to configure settings for ports on GigaVUE HC Series line cards and modules, including aliases, port filters, port parameters, and port types.

The **port** command has the following syntax:

```
port <port-id | port-alias | port-list>
  alarm
     buffer-threshold <0-100%> | [rx <0-100%> | tx <0-100%]>
    high-utilization-threshold <0-100%>
```

```
    low-utilization-threshold <0-100%>
  alias <alias string>
  assign role <user role> [level 1 | 2 | 3 | 4]
  buffer-index <<0-7> | low | default | high>
  comment <comment>
  egress-vlan strip
  filter rule
    add <drop <criteria>| pass <criteria>>
    delete <all | rule-id <rule ID>>
    edit rule-id <rule-ID> [drop <criteria> | pass <criteria>]
  header-strip <mpls-advanced | mpls-l3 | vxlan >
  ingress-vlan-tag <2-4000>
  lock [description <description>]
  lock-share <user <username>>
  mode <none | 4x10G | 4x25G | 4x100G | 2x40G>
  params
     admin <disable | enable>
     autoneg <disable | enable>
     brief [port-list <port list>]
     discovery <cdp | lldp | all | disable>
     duplex <full>
     fec <cl91|cl74|cl108 | off>
     forcelinkup <disable | enable>
     gdp <enable | disable>
     speed <10 | 100 | 1000 | 10000 | 25000 | 40000 | 100000>
     ude <enable | disable>
     taptx <active | passive>
  ptp <enable | disable>
        vlan <value>
     announce-interval <value>
     delay-req-interval <value>
     sync-interval <value>
 tag-protocol-id <TPID value>
  timestamp <append-ingress | source-id <0-65535> | strip-egress>
        threshold [rx | tx] [drop | error] [count | percent] [value <value>]
        timestamp
     ingress insert
     ingress source-id <value>
     egress insert
     egress source-id <value>
     ingress disable
     egress disable
  tool-share role <user role>
  type <hybrid | inline-network | inline-tool | network | stack | tool | circuit>
  l2gre-id <L2GRE identifier>
  vxlan-id <VXLAN identifier>
  zr-optics <add-tx-opt-pow | app-code | grid-frequency | grid-spacing | part-number>
```

The following table describes the arguments for the **port** command.

Argument	Description
port-id | port-alias | port-list	Specifies the ports on which to configure settings. Use one of the following: • **port-id, port-alias, port-list**—Specifies ports using the standard conventions described in Port Lists Definition in the GigaVUE-OS.
alarm buffer-threshold <0-100%> | [rx <0-100%> | tx <0-100%]>	Specifies buffer alarm thresholds on a port, as a percentage from 1 to 100. You can specify the alarm buffer threshold in the rx and tx directions for network and stack type ports and in the tx direction for tool type ports. The default is 0, which disables the threshold. For example: **(config) # port 2/1/x1 alarm buffer-threshold 50** **(config) # port 2/1/x1 alarm buffer-threshold rx 60**
high-utilization-threshold <0-100%>low-utilization-threshold <0-100%>	Specifies high and low utilization thresholds on a port as a percentage from 1 to 100. The utilization thresholds specify the value at which a utilization alarm will be generated for a port, either rising or falling. Alarms are reported to all configured SNMP trap destinations and recorded in the log file. The default is 0, which disables the threshold. For example: • Specifying high and low utilization thresholds for front-end ports: **(config) # port 1/1/x1 alarm high-utilization-threshold 70** **(config) # port 1/1/x1 alarm low-utilization-threshold 20** • Specifying high and low utilization thresholds for GigaSMART engine port: **(config) # port 1/1/e1 alarm high-utilization-threshold 70** **(config) # port 1/1/e1 alarm low-utilization-threshold 20**
alias <alias string>	Specifies an alias for a particular port. Aliases can be used in place of the numerical **bid/sid/pid** identifier required in many packet distribution commands in the GigaVUE-OS CLI. For example, instead of configuring a map from 1/1/x1 to 1/2/x4, you could create a map from **Gb_In** to **Stream-to-Disk**. For example: **(config) # port 24/1/x1 alias myPort** **NOTE:** Aliases can only be applied to single ports. They cannot be applied to groups of ports. To use an alias for a group of ports, use the tool-mirror feature.
assign role <user role> [level 1 | 2 | 3 | 4]	Specifies a required role that a user must be assigned in order to access the specified port. The access is granted at the specified permission level, as follows: • **Level 1**—Read-only access. Can view port configuration and statistics. • **Level 2**—Level 1 plus the capability to configure port-lock, lock-share, and all traffic objects except port-pair. • **Level 3**—Level 2 plus the capability to configure port params (such as administrative status of the port, speed, duplex, and auto-negotiation), as well as port-pair.

Argument	Description
	• **Level 4**—Level 3 plus the capability to change the port type. For example: **(config) # port 24/1/x1 assign role Default level 1**
buffer-index <<0-7> | low | default | high>	Configures a buffer-index value that allocates shared buffer space (memory) on specified ports to reduce oversubscription or to handle bursty traffic as follows: • **0-7**—Specifies a value for the buffer index. • **low**—Specifies a low buffer index, which is the same as a value of 3. • **default**—Specifies a default buffer index, which is the same as a value of 5. • **high**—Specifies a high buffer index, which is the same as a value of 7. Use this parameter on tool and hybrid ports to reduce drops when the specified ports are oversubscribed or when traffic on the ports is bursty. **NOTE:** Configuring the buffer-index value as low or default. Configuring a high buffer-index value causes packet drop in network ports (as there are no buffers for the incoming traffic). Use the **show buffer-index** command to display the configuration of the buffer index on ports. Examples: **(config) # port 1/1/x1 buffer-index low** **(config) # port 1/1/x2..x5 buffer-index 2**
comment <comment>	Specifies a unique text string that describes the port. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # port 24/1/x1 comment "My port"**
egress-vlan strip	Enables outer VLAN stripping on specified egress ports. The egress port type must be a tool or hybrid. Use the **show egress-vlantag** command to display the configuration of outer VLAN stripping on egress ports. Examples: **(config) # port 1/2/x1 egress-vlan strip** **(config) # port 1/2/x2..x3 egress-vlan strip** If a port is configured for egress VLAN stripping, the port type cannot be changed until egress VLAN stripping is disabled. To disable outer VLAN stripping on specified egress ports: **(config) # no port 1/2/x1 egress-vlan strip** **NOTE:** Port level egress VLAN stripping is only supported if from/source ports are not on the same node as tool/hyrbid port with egress-vlan strip in double tag mode on GigaVUE-HC1-Plus, GigaVUE-TA25, and GigaVUE-TA25E. Refer to *"Configuring Egress Port VLAN Stripping"* section in the *GigaVUE Fabric Management Guide* for details.

Argument	Description
filter rule add <drop <criteria> | pass <criteria>> delete <all | rule-id <rule ID>> edit rule-id <rule-ID> [drop <criteria> | pass <criteria>]	Manages filters on the specified Egress port. Note that the **filter** argument is supported for tool, hybrid, circuit, and inline network ports. Network ports use maps to direct traffic. **NOTE:** Port filters for inline network ports are not supported on GigaVUE TA Series devices. **NOTE:** In the case of inline network LAG, the port filters must be applied on each of the inline network ports that are part of the inline network LAG. The criteria available for egress port filters is mostly the same as that used for filters—you can create pass or drop filters based on the same packet criteria available for maps. You can see the available criteria in the CLI by typing a command similar to the following: **(config) # port <egress port number> filter rule add pass ?** Refer to the *Port-Filters* section in the *GigaVUE Fabric Management Guide* for a description of the available filter criteria. **NOTE:** You can configure egress port filters on a single port simultaneously. The filter argument is blocked when using the **port** command with multiple egress ports or port groups. Examples: The following tool port filter drops all packets with a VLAN ID between 100..200 from tool port 14/2/g40: **(config) # port 14/2/g40 filter rule add drop vlan 100..200** The following command passes only IPv6 traffic on tool port 14/2/g44: **(config) # port 14/2/g44 filter rule add pass ipver 6** Use the **# show filter-resource** command to view the maximum filter resources available and the filter resources used for a device. **NOTE:** Egress port filters are supported on GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-HCT, and GigaVUE-HC1-Plus, except that VLAN rules are not supported with port filters, and Either IPv4 or IPv6 type port filter rules are supported only if L2 circuit encapsulation tunnels or GigaSMART maps are used else both IPv4 and IPv6 rules are supported.
ingress-vlan-tag <2-4000>	Configures an ingress port VLAN tag, which is a number from 2 to 4000 that is added to a packet. The port must be a network or inline-network, or hybrid type of port. For example: **(config) # port 7/1/x1 ingress-vlan-tag 100** **Note**: On GigaVUE-TA25 and GigaVUE-TA25E, ingress VLAN tagging is not supported on network ports associated with GSOP maps, and GSOP maps are not supported on network ports configured with ingress VLAN tag. This limitation is not applicable to the E-tag mode.
lock [description <description>]	Restricts use of the specified port for only your user account as follows: • Users with the admin role can lock any port in the system. Users with the Default/Operator role assigned can only lock ports to which their account has been

Argument	Description
	granted access. • Administrators can lock a port for another user by including the optional user <username> argument. • You can optionally share a locked port with the lock-share argument. Refer to the *Setting Locks and Lock-Shares* section in the *GigaVUE Administration Guide* for details on how locks are used. For example: **(config) # port 7/1/x1 lock**
lock-share <user <username>>	Shares a locked port with another user account. For example: **(config) # port 7/1/x1 lock-share user operator**
mode <none | 4x10G | 4x25G | 4x100G | 2x40G>	Configures the port breakout mode as follows: • **none**—Specifies no port breakout mode. This is the default mode for GigaVUE-OS nodes. • **4x10G**—Specifies the **4x10Gb** port breakout mode. This mode provides a 4 x 10Gb breakout option for 100Gb/40Gb ports. The **4x10G** mode only applies to GigaVUE-TA100, GigaVUE-TA200, GigaVUE-TA25, the PRT-HC3-C08Q08, PRT-HC3-C16, SMT-HC3-C05, and BPS-HC3-C25F2G modules on GigaVUE-HC3, **NOTE:** Starting in software version 5.5, GigaVUE-TA40 supports 4x10Gb breakout at port level. Port breakout mode in GigaVUE-TA40 is configured as follows: o 24 out of the 32 ports provide 4x10Gb breakout support. The first 12 ports and the last 12 ports provide support for breakout functionality with 96 sub-ports operating as 10Gb ports ■ Ports q1 to q12 and q21 to q32 support breakout functionality ■ Ports q13 to q20 do not support breakout functionality ■ Ports are named as q1x1....q1x4, q2x1...q2x4 (similar to other hardware platforms) to support the breakout functionality • **4x25G**—Specifies the **4x25G** port breakout mode. This mode provides a 4 x 25Gb breakout option for 100Gb QSFP28 SR ports. The **4x25G** mode only applies to GigaVUE-TA200 , GigaVUE-TA25, and the PRT-HC3-C08Q08, PRT-HC3-C16, and SMT-HC3-C05 modules on GigaVUE-HC3. • **4x100G**—Specifies the **4x100G** port breakout mode. This mode supports 4x100G for TA400, and ports are e.g. 1/1/d1c1 • **2x40G**—Specifies the **2x40G** port breakout mode. This mode provides a 2x40Gb breakout option for 100Gb/40Gb ports. The **2x40G** mode only applies to the PRT-HC3-C08Q08 module on GigaVUE-HC3. **For the BPS-HC3-C25F2G module on GigaVUE-HC3, refer to the *GigaVUE-HC3 Hardware Installation Guide.*** **NOTE:** When there is a port breakout mode, the port command for the breakout port cannot be used with a ".. " notation. For Example: To admin enable the port breakout mode:

Argument	Description
	(config) # port 1/1/c4x1 params admin enable **(config) # port 1/1/c4x2 params admin enable**
	The 100Gb ports that support **4x10Gb** mode can operate at 40Gb speed with QSFP+ SR or PLR4 transceivers. When a parent port is configured in **4x10Gb**, it can be broken out into four 10Gb ports, called subports. The subports will all have the same speed (10Gb). Subports have x1 to x4 appended to their port ID, for example, 1/1/c2x1. The 100Gb ports that support **4x25Gb** mode can be broken out into four times 25Gb ports, called subports. The subports will all have the same speed (25Gb). Subports have x1 to x4 appended to their port ID, for example, 1/1/c2x1. The 100Gb ports that support **2x40Gb** mode can operate at 40Gb speed with QSFP+ SR and LR transceivers. When a parent port is configured in **2x40Gb** mode, it can be broken out into two 40Gb ports, called subports. The subports will all have the same speed (40Gb). Subports will have q1 to q2 appended to their port ID, for example, 1/1/c1q1 and 1/1/c1q2. The subports in the PRT-HC3-C08Q08 module on GigaVUE-HC3 Control Card version 1 (CCv1) that function as stack ports must be of the same port type.
	In general, subports created from port breakout modes can function as network, tool, or hybrid ports, as well as GigaStream port members, but they cannot function as stack ports. However, 10Gb stacking is supported only on GigaVUE-TA25GigaVUE-TA100, and PRT-HC3-C08Q08 on GigaVUE-HC3 when ports are broken out into **4x10Gb** mode. Use breakout cables or breakout panels (PNL-M341 or PNL-M343). Refer to the respective *Hardware Installation Guide*. The default mode is **none**. **NOTE:** On GigaVUE-TA25, breakout subports support all port types. Only C1, C5 ports support 4x10Gb, 4x25Gb breakout modes. **NOTE:** Each port can only have one mode. Examples of configuring parent ports and subports are listed on the next page.
mode <none | 4x10G | 4x25G | 4x100G | 2x40G> (continued)	The following is an example of **4x10Gb** mode: **(config) # port 1/1/c2 mode 4x10Gb** In this example, the subports will be 1/1/c2x1, 1/1/c2x2, 1/1/c2x3, and 1/1/c2x4. Once the 1/1/c2 port (the parent port) is in the **4x10Gb** mode, it is no longer available to be used in any configuration. For example if 1/1/c2 is used in a map, it will be rejected as an invalid port. Refer to the next page for rules for configuring parent ports and subports. When a parent port, for example, 1/1/c2, is broken out, you cannot specify, for example, 1/1/c1..c4 in a port list, since 1/1/c2 will be rejected as an invalid port. To specify all the ports from 1/1/c1 to 1/1/c4 in a port list, including the subports, use the syntax as follows: **(config) # show port params port-list 1/1/c1,1/1/c2x1..c2x4,1/1/c3..c4** Below are examples of valid port lists when 1/1/c2 is broken out: **1/1/c2x1..c2x4** includes **1/1/c2x1, 1/1/c2x2, 1/1/c2x3, 1/1/c2x4,** **1/1/c2x2..c4** includes **1/1/c2x2, 1/1/c2x3, 1/1/c2x4** **1/1/c3, 1/1/c41/1/c1..c2x3** includes **1/1/c1, 1/1/c2x1, 1/1/c2x2, 1/1/c2x3**

Argument	Description

	Below are examples of invalid port lists when 1/1/c2 is broken out:
	1/1/c2..c2x3, 1/1/c1..c3
	The output of **show** commands will also display invalid ports. For example, when 1/1/c2 is broken out, but 1/1/c1 and 1/1/c3 are not, the following is displayed:
	(config) # show port params port-list 1/1/c1..c3 brief
	`Port Type Alias Admin ---------- ------1/1/c2 network - disabled ! Invalid port 1/1/c31/1/c4 network - disabled`
	The following is a **2x40Gb** mode example:
	(config) # port 1/1/c3 mode 2x40Gb
	In this example, the subports will be: 1/1/c3q1 and 1/1/c3q2.
	(config) # show port params port-list 1/1/c3q1..c3q2 brief
	`Port Type Alias Admin ---------- ------1/1/c3q1 network - disabled 1/1/c3q2 network - disabled`
	To change the port mode back to **none**, either clear the mode or set it to **none**, as follows:
	(config) # no port 1/1/c2 mode
	or
	(config) # port 1/1/c2 mode none
	The mode can be changed from **2x40Gb** to **4x10Gb** or from **4x10Gb** to **2x40Gb** without switching to **none**.
	Rules for configuring parent ports and subports are listed on the next page.

mode <none | 4x10G | 4x25G | 4x100G | 2x40G> (continued)	The following are rules for configuring parent ports and subports: • You can change the port mode from **none** to **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** if the parent port has port-level configuration (any of the parameters under the **port** command). However, you cannot change the port mode from **none** to **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** if there is an egress port filter configured on the parent port. Remove the filter first, then change the mode. • You can change the port mode from **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** to **none** if the subports have port-level configuration (any of the parameters under the **port** command). However, you cannot change the port mode from **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** to **none** if there is an egress port filter configured on the subport. Remove the filter first, then change the mode. • You cannot change the port mode from **none** to **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** if there is any traffic configuration on the parent port such as map, map-passall, map-scollector, port-pair, or tool-mirror. Remove the traffic configuration first, then change the mode. • You cannot change the port mode from **4x10Gb** or **2x40Gb** or **4x25Gb** or **4x100Gb** to **none** if there is any traffic configuration on the subport such as map, map-passall, map-scollector, port-pair, or tool-mirror. Remove the traffic configuration first, then change the mode. • You cannot configure any parameters on a parent port if the port is broken out into subports. For example, if you try to configure port 1/1/c1 when it has subports 1/1/c1x1..c1/x4, the following error message is displayed:

Argument		Description
		(config) # port 1/1/c1 type tool% Invalid port '1/1/c1'. • You cannot configure any parameters on a subport if the port is not broken out into subports. For example, if you try to configure port 1/1/c2x1 when 1/1/c2 is not broken out, the following error message is displayed: **(config) # port 1/1/c2x1 type tool% Invalid port '1/1/c2x1'.** • In general, subports cannot be configured as stack ports. An error message is displayed as follows: **(config) # port 1/1/c1x4 type stack% Port 1/1/c1x4 is not parent port and does not support type stack** or **(config) # port 5/4/c3q1 type stack** **% Port 5/4/c3q1 is not parent port and does not support type stack** • QSFP+ LR transceivers do not support **4x10Gb** port breakout. If you try to breakout a port with QSFP+ LR transceiver, an error message is displayed as follows: **% Port 1/1/c3 does not support breakout mode** • For the BPS-HC3-C25F2G module on GigaVUE-HC3, if the default inline network (default_inline_net) is modified, you cannot change the port mode from **none** to **4x10Gb** or from **4x10Gb** to **none**. An error message is displayed as follows: **(config) # port 2/2/c1 mode none% Port '2/2/c1x1' is being used by:1) inline-network default_inline_net_2_2_1_1 (not default values)**
params		Configures port administration and physical parameters.
	admin <disable | enable>	Administratively enables or disables ports. Use the **show port params** command to see the administrative status of ports. For example: **(config) # port 7/1/x1 params admin enable**
	autoneg <disable | enable>	Enables or disables auto-negotiation for a port. When auto-negotiation is enabled, duplex and speed settings are ignored. **Autonegotiation Limitations:** o Autonegotiation is always disabled for for 10Gb, 25Gb, 40Gb, and 100Gb ports. o Auto-negotiation must be enabled for 1Gb speeds over copper, per the IEEE 802.3 specification. o Auto-negotiation is not supported for 1Gb fiber ports on Gigamon Platforms. Examples: **(config) # port 7/1/x1 params autoneg disable** **(config) # port 1/2/g7 params autoneg enable**
	discovery <cdp | lldp | all |	Configures port discovery options on network, tool, or circuit type ports, as follows: • **cpd**—Enables CDP port discovery. • **lldp**—Enables LLDP port discovery. • **all**—Enables both CDP and LLDP port discovery protocols.

Argument		Description
	disable>	• **disable**—Disables port discovery. For example: **(config) # port 7/1/x1 params discovery cdp** Discovery is disabled by default. To enable discovery, configure one or more protocols. Refer to the "*Port Discovery with LLDP and CDP*" section in the GigaVUE Fabric Management Guide.
	duplex <full>	Specifies the port's duplex configuration. Only full-duplex is supported. Half duplex support is removed from all GigaVUE-OS nodes. If half duplex was configured in a previous software version, it will remain intact following the upgrade to 5.2 or higher release. Update to full duplex, if required. For example: **(config) # port 7/1/x1 params duplex full**
	fec <cl91| cl74|cl108| off>	Configures forward error correction (FEC) on the port to ensure error-free traffic over long distances. The values are: • cl91—Enables FEC on the port and supports 25Gb, 100Gb, and 4x25Gb speeds. • cl74— Enables FEC on the port and supports 25Gb, and 4x25Gb speeds. • cl108 — Enables FEC on the port and supports 25Gb speed only on GigaVUE-TA25 and GigaVUE-TA25E. • off—Disables FEC on the port and supports 25Gb, 100Gb, and 4x25Gb speeds. For example: **(config) # port 7/1/x1 params fec cl91**
	forcelinkup <disable | enable>	Forces connection on an optical port. Use this option when an optical tool port is connected to a legacy optical tool that does not transmit light. **Forcelinkup** is available as follows: • Available for tool ports. • Available for optical 40/100/400Gb network ports. • Not available for 10Gb-capable ports with a 1Gb SFP installed. 10Gb-capable optical tool ports only support **forcelinkup** when a 10Gb SFP+ is installed. • Not available on 100Gb ports with CFP2 transceivers. Use **forcelinkup** on a tool port and optical 40/100/400Gb network ports. The forcelinkup status will be **on** when: • the tool port and network port are administratively enabled. • **forcelinkup** is enabled, with or without a transceiver or cable for the tool port. • the tool port link is up. • **forcelinkup** is enabled, with a transceiver or cable for the network port. • the network port link is up for 40/100/400Gb links. For example: **(config) # port 7/1/x1 params forcelinkup enable**
	gdp	Enables or disables the Gigamon discovery (GDP) on the port.

Argument		Description
	<enable | disable>	• **enable**—Enables Gigamon discovery (GDP) on the port. • **disable**—Disables Gigamon discovery (GDP) on the port. Gigamondiscovery is disabled by default on the port. Gigamon discovery cannot be enabled on inline-tool and inline-network type ports.
	speed <10 | 100 | 1000 | 10000 | 25000 | 40000 | 100000>	Sets the line speed of a port as follows: • **10**—10Mbps • **100**—100Mbps • **1000**—1Gbps • **10000**—10Gbps • **25000**—25Gbps • **40000**—40Gbps • **100000**—100Gbps Sets the port speed in Mbps if auto-negotiation is off. Examples: **(config) # port 1/1/x1 params speed 10000** **(config) # port 7/1/c1 params speed 40000** Use **40000** and **100000** to change the port speed of each inline network port pair (c1/c2 or c3/c4) on the BPS-HC3-C25F2G module on GigaVUE-HC3. You only need to configure one port in each pair. Use **25000** to change the port speed on GigaVUE-TA25E, GigaVUE-HC1-Plus, GigaVUE-TA200 and on the following GigaVUE-HC3 modules: • PRT-HC3-X24 • BPS-HC3-C25F2G • BPS-HC3-Q35C2G • BPS-HC3-C35C2G • PRT-HC1P-Q04X08 • HC1P-C04X08 **NOTE:** When AOC dual rate optics are used, the Link is down after changing the port speed from 40Gb to 100Gb between GigaVUE-HC3V2, PRT-HC3-C08, and GigaVUE-TA25, GigaVUE-TA25Edevices. To resolve this problem, first enable 40Gb speed in the GigaVUE-TA25 and GigaVUE-TA25E devices, then configure speed in the GigaVUE-HC3 device.
	ude <enable | disable>	Indicates whether the port is enabled for unidirectional (Ude) or bidirectional traffic. Enabled means unidirectional; Disabled means bidirectional. **NOTE:** Do not disable UDE on Out Ports when using BiDi RX/TX optics connecting towards passive TAPs. For example, if BiDi RX/TX optics are connected to both the Tap-facing and Network-facing ports of the TAP-M506/506T, disabling UDE on the TAP-facing ports will cause a production link failure. For example, you can enable the port and disable unidirectional ethernet in a single line or separately:

Argument		Description
		(config) # port 7/1/x1 params admin enable ude disable -or- **(config) # port 7/1/x1 params admin enable** **(config) # port 7/1/x1 params ude disable**
	taptx <active | passive>	Opens or closes the copper TAP port relay. Active closes the port relay. Passive opens the port relay. For example: **(config) # port 7/1/x1 params taptx active**
ptp	**<enable | disable>**	Enables or disables PTP configuration on a port. For example: **(config) # port 1/1/c1 ptp enable** **(config) # port 1/1/c1 ptp disable**
	vlan <value>	Configures the PTP Port VLAN wihin the range 2-4000. For example: **(config) # port 1/1/x1 ptp vlan <35>** Use the command **show vlan-resource** to identify available VLAN resources. Refer to show section for more information.
	announce-interval <value>	Configures the time interval between PTP announce messages in the port in seconds on a logarithmic scale. • For Domain 0, the valid range is between -2 and 4 log seconds. The default is 1. • For Domain 24-43, the valid range is between -3 and 4 log seconds. The default is -3. For example: **(config) # port 1/1/c1 ptp announce-interval -3**
	delay-req-interval <value>	Configures the minimum interval allowed between PTP delay request messages. The valid range is between -7 and 0 log seconds. • For Domain 0, the default is -5. • For Domain 24-43, the default is -4. For example: **(config) # port 1/1/c1 ptp delay-req-interval -4**
	sync-interval <value>	Configures the time interval between PTP synchronization messages on the port. The valid range is between -7 and 0 log seconds. • For Domain 0, the default is -5. • For Domain 24-43, the default is -4. For example: **(config) # port 1/1/c1 ptp sync-interval -4**
threshol	**<rx> [drop |**	Configures drop or error threshold parameters in the rx direction.

Argument		Description
d	**error] [count | percent] value <value>**	
	<tx> [drop | error] [count | percent] value <value>	Configures drop or error threshold parameters in the tx direction.
tag-protocol-id <TPID value>		Configures a TPID value for the ingress port VLAN tag. The default value of TPID is 0x8100, and other supported values are 0x9100 and 0x88a8. • If tag-protocol-id is configured without ingress-VLAN-tag on the port, the CLI displays the following error message. "Ingress Vlan Tagging is not enabled on Port <port>; Configure ingress-vlan-tag before configuring tag-protocol-id" • Upon unconfiguring ingress-vlan-tag on a port, the tag-protocol-id configured on the respective port would be unconfigured (set to the default value: 0x8100). For example: **(config) # port 1/1/x3 tag-protocol-id 0x88a8**
timestamp <append-ingress | source-id <0-65535> | strip-egress>		Configures timestamping options. For example: **(config) # port 7/1/x1 timestamp append-ingress** **(config) # port 7/1/x1 timestamp source-id 500**
timestamp	**ingress insert**	Configures ingress timestamp on a port. **NOTE:** Use these arguments for GigaVUE-TA200 devices. **NOTE:** Use the show timestamp brief command to display the PTP based timestamp information. For example:

Argument		Description
		(config) # port 1/1/c3 timestamp ingress insert
	ingress source-id <value>	Specifies a unique ID for the ingress timestamp. The packets that ingress the port will be timestamped with this unique source ID. The source ID enables you to identify the port on which the timestamp is applied. For example: **(config) # port 1/1/c3 timestamp ingress source-id 113**
	egress insert	Configures egress timestamp on a port. For example: **(config) # port 1/1/c2 timestamp egress insert**
	egress source-id <value>	Specifies a unique ID for the egress timestamp. The packets that egress the port will be timestamped with this unique source ID. The source ID enables you to identify the port on which the timestamp is applied. For example: **(config) # port 1/1/c2 timestamp ingress source-id 112**
	ingress disable	Disables the ingress timestamp on the port. For example: **(config) # port 1/1/c3 timestamp ingress disable**
	egress disable	Disables the egress timestamp on the port. For example: **(config) # port 1/1/c2 timestamp egress disable**
tool-share role <user role>		Designates a tool port as available for tool-to-tool pass-alls (tool-mirrors) with the specified roles. For example: **(config) # port 7/1/x1 tool-share role admin**
type <hybrid | inline-network | inline-tool | network | stack | tool | circuit>		Specifies a port as a hybrid, inline-network (input), inline-tool (output), network (input), tool (output), stack, or circuit port. For example, the following command configures port g2 in slot 2 on box 1 as a tool port: **(config) # port 1/2/g2 type tool** All ports on GigaVUE HC Series line cards or modules are available as network and tool ports. Stack ports are supported at speeds of 10Gb, 40Gb, and 100Gb. However, stack ports are not supported on SFP-531 optics (10GBASE-T) and SFP+ 531T optics. A hybrid port is a physical port that has a dual function as a network port and as a tool port. Refer to the *"Working with Hybrid Ports"* section in the *GigaVUE Fabric Management Guide*. The inline-network and inline-tool type of ports can only be configured on GigaVUE HC Series nodes. Refer to the *"Configuring Inline Bypass Solutions"* chapter in the *GigaVUE Fabric Management Guide*. **Circuit ports are used to send or receive traffic that is tagged with a circuit ID. For more information, refer to the *"Circuit Ports"* section in the *GigaVUE Fabric Management Guide*.**

Argument	Description
l2gre-id <L2GRE identifier>	Configures the L2GRE ID that is added to the L2GRE group created for a specific device . The ID configured for the network port overrides the ID configured at the chassis-level. For details about the CLI command used to configure the L2GRE group, refer to Configure L2GRE Group. For example: **(config) # port 7/1/x1 l2gre-id 400** **NOTE:** You cannot specify both L2GRE ID and VXLAN ID for a network port.
vxlan-id <VXLAN identifier>	Configures the VXLAN ID that is added to the VXLAN group created for a specific device. The ID configured for the network port overrides the ID configured at the chassis level. For details about the CLI command used to configure the VXLAN group, refer to Configure VXLAN Group. For example: **(config) # port 7/1/x1 vxlan-id 300** **NOTE:** You cannot specify L2GRE ID and VXLAN ID for a network port.
header-strip <vxlan | mpls-l3| mpls-advanced>	Enables the Header Stripping functionality on the specified network or hybrid port. For example: **(config) # port 1/1/c1 header-strip vxlan** **(config) # port 1/1/c1 header-strip mpls-l3** **NOTE:** You cannot enable VXLAN and MPLS Header Stripping on the same port. For more information, refer to the *"VXLAN Header Stripping – Rules and Notes"* and *"MPLS Header Stripping – Rules and Notes"* sections in the *GigaVUE-FM User's Guide*.
header-strip **vxlan**	Configure the port to strip the VXLAN header from incoming packets. For example: **(config) # port 1/1/c1 header-strip vxlan**
mpls-l3	Configure the port to strip the MPLS Layer 3 header from incoming packets. It can only be used in networks where there is only L3 MPLS traffic. For example: **(config) # port 1/1/c1 header-strip mpls-l3**
mpls-advanced <ip4-wo-options | ip4-options | ipv6 | l2-wo-cw-ach | l2-	Configure the port for advanced MPLS Header Stripping, which supports more complex packet formats. This feature is exclusively available for the GigaVUE-TA400 device. Allows user to configure multiple options based on MPLS traffic types present in network. Refer to the *Advanced MPLS Header Stripping for GigaVUE-TA400* section in the *GigaVUE Fabric Management Guide* for a description of the advanced MPLS Header Stripping.

Argument	Description
cw-ach>	
ip4-wo-options	Configure the port to strip the MPLS Header from the L3 MPLS IPv4 (without options) packets. Enable this option only if L3 MPLS IPv4 (without options) traffic type is present in network. IPv4 without options has a standard 20-byte header for efficient routing. For example: **(config) # port 1/1/d1 header-strip mpls-advanced ip4-wo-options**
ip4-options	Configure the port to strip MPLS Header from the L3 MPLS IPv4 (with options) packets. Enable this option only if L3 MPLS IPv4 (with options) traffic type is present in network. IPv4 with options includes additional fields (up to 60 bytes) for advanced functions like time stamping. For example: **(config) # port 1/1/d1 header-strip mpls-advanced ip4-options**
ipv6	Configure the port to strip the MPLS header from the L3 MPLS IPv6 packets. Enable this option only if L3 MPLS IPv6 traffic type is present in network. The standard IPv6 header (40 bytes) follows the MPLS label stack and contains source and destination IPv6 addresses, flow label, and other fields. For example: **(config) # port 1/1/d1 header-strip mpls-advanced ip6**
l2-wo-cw-ach	Configure the port to strip the outer Ethernet and MPLS headers from the incoming L2 MPLS without Control Word(CW) or Associated Channel Header(ACH) packets. Enable this option only if L2 MPLS (without CW/ACH) traffic type is present in network. L2 MPLS without CW or ACH provides a basic MPLS label-switched path for transporting Layer 2 packets without the extra control, sequencing, or signaling functionalities provided by the CW and ACH headers. For example: **(config) # port 1/1/d1 header-strip mpls-advanced l2-wo-cw-ach**
l2-cw-ach	Configure the port to strip the outer Ethernet and MPLS headers along with Control Word (CW) or Associated Channel Header(ACH) from the incoming L2 MPLS with CW or ACH packets. Enable this option only if L2 MPLS (with CW/ACH) traffic type is present in network. L2 MPLS with CW ensures packet sequencing and integrity for Layer 2 frames, while ACH enables Operations, Administration, and Maintenance (OAM) transport for network monitoring and management, especially in MPLS traffic. For example: **(config) # port 1/1/d1 header-strip mpls-advanced l2-cw-ach**
zr-optics <add-tx-opt-pow | app-code | grid-frequency | grid-spacing | part-	The zr-optics command configures the interoperability between Innolight 400Gb and Cisco ZR/ZR+ optics when connected to a GigaVUE-TA400 and paired with Cisco ZR/ZR+ optics in a Cisco Router under OpenZR compliance. It manages the necessary low-level parameter tuning required for connection stability and performance. **NOTE:** Avoid hot-swapping, reloading, or power cycling while executing configuration commands, as this may interrupt the firmware update process.

Argument		Description
number>		For example: **(config) # port 1/1/d1 zr-optics zr-grid-spacing 6250 zr-grid-frequency 192000 zr-add-tx-opt-pow -2.5 zr-app-code 5 part-number DP04QSDD-HE0-19**
zr-optics	**add-tx-opt-pow**	Configures the ZR+ optics transmit power offset in increments of 0.5 dBm. ZR+ optics transmit power ranges from -5 to +5 dBm in increments of 0.5. For example: **(config) # port 1/1/d1 params zr-add-tx-opt-pwr 2.5**
	app-code	Configure the ZR+ application select code, with options ranging from 1 to 8. ZR+ application select code ranges from 1 to 8. It is applicable only for Cisco ZR/ZR+ optics. For example: **(config) # port 1/1/d1 params zr-appselcode 1**
	grid-frequency	Configures the ZR+ grid frequency in GHz, ensuring alignment between devices. ZR+ grid frequency ranges from 191300 to 196200 GHz in increments of 100. The default value is 193100 GHz. For example: **(config) # port 1/1/d1 params zr-frequency 192000**
	grid-spacing	Configure the ZR+ grid spacing in MHz, which must match between interconnected optics. ZR+ grid spacing ranges from 3125, 6250, 12500, 25000, 33000, 50000, 75000, and 100000 MHz. The 33,000 MHz ZR+ grid spacing is supported only on Innolight optical transceivers. The default value is 100000 MHz. For example: **(config) # port 1/1/d1 params zr-grid-spacing 6250**
	part-number	Configure the part number for ZR/ZR+ optics, which is required for offline provisioning. The part numbers of the ZR/ZR+ optics are listed below. • Cisco ZR/ZR+ Optics: DP04QSDD-HE0-19 • Innolight ZR+ Optics: DP04QSDD-HE0-19 • Innolight ZR Optics: GMONDS400SZRTA0 For example: **(config) # port 1/1/d1 zr-optics part-number DP04QSDD-HE0-19**

Related Commands

The following table summarizes other commands related to the **port** command:

Task	Command
Displays all port parameters in table format.	**# show port**
Displays all port access.	**# show port access all**
Displays port access for a specified box.	**# show port access box-id 2**
Displays port access for a specified box in table format.	**# show port access box-id 2 brief**
Displays port access for a specified port list.	**# show port access port-list 2/1/x1..x2**
Displays port access for a specified port list in table format.	**# show port access port-list 2/1/x1..x4 brief**
Displays port access for a specified slot.	**# show port access slot 1**
Displays port access for a specified slot in table format.	**# show port access slot 1 brief**
Displays all port aliases.	**# show port alias**
Displays all port assignments.	**# show port assignment all**
Displays port assignments for a specified box.	**# show port assignment box-id 3**
Displays port assignments for a specified box in table format.	**# show port assignment box-id 3 brief**
Displays port assignments for a specified port list.	**# show port assignment port-list 3/8x1**
Displays port assignments for a specified port list in table format.	**# show port assignment port-list 3/8x1 brief**
Displays port assignments for a specified slot.	**# show port assignment slot 3/8**
Displays port assignments for a specified slot in table format.	**# show port assignment slot 3/8 brief**
Displays all port comments.	**# show port comment**
Displays port discovery information.	**# show port discovery**
Displays port discovery information for a specified box.	**# show port discovery box-id 2**
Displays port discovery information for a specified box in table format.	**# show port discovery box-id 2 brief**
Displays port discovery information in table format.	**# show port discovery brief**
Displays port discovery information in table format for a specified port list.	**# show port discovery brief port-list 2/1/x1**
Displays port discovery information for specified port lists.	**# show port discovery port-list 2/1/x1..x4**
Displays port discovery information for specified port lists in	**# show port discovery port-list**

Task	Command
table format.	**2/1/x1..x4 brief**
Displays port discovery information for the specified slot.	**# show port discovery slot 4**
Displays port discovery information for the specified slot in table format.	**# show port discovery slot 4 brief**
Displays faceplate port number mapping to GigaVUE-OS port number.	**# show port faceplate-number-mapping**
Displays faceplate port number mapping to GigaVUE-OS port number for a specified port list.	**# show port faceplate-number-mapping port-list 2/1/x1**
Displays all port filters.	**# show port filter**
Displays port filters for a specified port list.	**# show port filter port-list 2/1/x1**
Displays port filter statistics for all line cards	**# show port filter stats all**
Displays port filter statistics for a line card placed inside the specified box.	**# show port filter stats box-id 1**
Displays port filter statistics for the specified list of ports.	**# show port filter stats port-list 1/1/x1..x6**
Displays port filter statistics for a line card placed at the specified slot.	**# show port filter stats slot 1**
Displays all ports that are in use by any maps (first-level, second-level maps) in table format.	**# show port in-use**
Displays all ports that are in use by any maps for the specified port type: network, tool, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port in-use type hybrid**
Displays the port mode on all ports of GigaVUE-OS nodes that support port mode.	**# show port mode**
Displays the port mode for the specified box.	**# show port mode box-id 1**
Displays the port mode for the specified slot.	**# show port mode slot 1**
Displays all port parameters.	**# show port params**
Displays all port parameters.	**# show port params all**
Displays port parameters for a specified box.	**# show port params box-id 2**
Displays port parameters for a specified box in table format.	**# show port params box-id 2 brief**
Displays brief port parameter information on the specified box ID for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine	**# show port params box-id 2 brief type stack**

Task	Command
port.	
Displays all port parameters that are in use by any maps for the specified box.	**# show port params box-id 2 in-use**
Displays all port parameters that are in use by any maps for the specified box, in table format.	**# show port params box-id 2 in-use brief**
Displays all port parameters that are in use by any maps for the specified box, in table format, and for the specified port type: network, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params box-id 2 in-use brief type tool**
Displays all port parameters that are in use by any maps for the specified box and for the specified port type: network, tool, hybrid, circuit, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params box-id 2 in-use type tool**
Displays port parameter information on the specified box ID for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params box-id 2 type stack**
Displays all port parameters that are not in use by any maps for the specified box.	**# show port params box-id 2 unused**
Displays all port parameters that are not in use by any maps for the specified box, in table format.	**# show port params box-id 2 unused brief**
Displays all port parameters that are not in use by any maps for the specified box, in table format, and for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params box-id 2 unused brief type hybrid**
Displays all port parameters that are not in use by any maps for the specified box and for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params box-id 2 unused type hybrid**
Displays all port parameters in table format.	**# show port params brief**
Displays port parameters in table format for a specified port list.	**# show port params brief port-list 2/1/x1..x2**
Displays all port parameters that are in use by any maps.	**# show port params in-use**
Displays all port parameters that are in use by any maps for the specified port type: network, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params in-use type inline-network**
Displays all port parameters for a specified port list. It also displays the ports and sub-ports in the specified range.	**# show port params port-list 2/1/x1..x4**

Task	Command
Displays all port parameters for a specified port list in table format. It also displays the ports and sub-ports in the specified range.	**# show port params port-list 2/1/x1..x4 brief**
Displays all port parameters for a specified slot.	**# show port params slot 3**
Displays all port parameters for a specified slot in table format.	**# show port params slot 3 brief**
Displays brief port parameter information on the specified slot ID for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 brief type gs**
Displays all port parameters that are in use by any maps for the specified slot.	**# show port params slot 3 in-use**
Displays all port parameters that are in use by any maps for the specified slot, in table format.	**# show port params slot 3 in-use brief**
Displays all port parameters that are in use by any maps for the specified slot, in table format, and for the specified port type: network, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 in-use brief type network**
Displays all port parameters that are in use by any maps for the specified slot and for the specified port type: network, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 in-use type network**
Displays port parameter information on the specified slot ID for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 type gs**
Displays all port parameters that are not in use by any maps for the specified slot.	**# show port params slot 3 unused**
Displays all port parameters that are not in use by any maps for the specified slot, in table format.	**# show port params slot 3 unused brief**
Displays all port parameters that are not in use by any maps for the specified slot, in table format, and for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 unused brief type network**
Displays all port parameters that are not in use by any maps for the specified slot and for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params slot 3 unused type stack**
Displays port parameter information for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or	**# show port params type tool**

Task	Command
gs for GigaSMART engine port.	
Displays all port parameters that are not in use by any maps.	**# show port params unused**
Displays all port parameters that are not in use by any maps for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port params unused type inline-tool**
Displays GigaVUE-OS port number mapping to faceplate port number.	**# show port port-number-mapping**
Displays GigaVUE-OS port number mapping to faceplate port number for a specified port list.	**# show port port-number-mapping port-list 2/1/x1**
Displays all port statistics.	**# show port stats all**
Displays port statistics for a specified box.	**# show port stats box-id 2**
Displays port statistics for a specified port list.	**# show port stats port-list inNet1**
Displays port statistics for a specified slot.	**# show port stats slot 1**
Displays share-tool-mirror roles for all ports.	**# show port tool-share all**
Displays share-tool-mirror roles for a specified box.	**# show port tool-share box-id 2**
Displays share-tool-mirror roles for a specified port list.	**# show port tool-share port-list 2/1/x1..x4**
Displays share-tool-mirror roles for a specified slot.	**# show port tool-share slot 1**
Displays port information for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port type network**
Displays all ports that are not in use by any maps (first-level, second-level maps), in table format.	**# show port unused**
Displays all ports that are not in use by any maps for the specified port type: network, stack, tool, circuit, hybrid, inline-network, inline-tool, or gs for GigaSMART engine port.	**# show port unused type tool**
Displays port utilization for all line cards or modules.	**# show port utilization all**
Displays port utilization for a specified box.	**# show port utilization box-id 2**
Displays port utilization for a specified port list.	**# show port utilization port-list 2/1/x1..x4**
Displays port utilization for a specified slot.	**# show port utilization slot 3**
Displays the L2GRE ID configured for the network port.	**# show port l2gre**

Task	Command
Displays the VXLAN ID configured for the network port.	**# show port vxlan**
Displays buffer profile current information.	**# show profile current buffer all**
Displays a minute of buffer profile history information.	**# show profile history buffer 2/1/x1 min**
Displays ports with modified buffer index.	**# show buffer-index**
Displays buffer usage by box ID.	**# show buffer box-id 2**
Displays buffer usage by port ID.	**# show buffer port 2/1/x1**
Displays buffer usage by port ID and direction.	**# show buffer port 2/1/x1 rx**
Displays buffer usage by slot.	**# show buffer slot 1**
Displays all ingress port VLAN tags.	**# show ingress-vlan-tag**
Displays the list of ports on which the Header Stripping protocol is enabled for a box ID.	**# show port header-strip box-id 1**
Displays the details of the Header Stripping protocol that is enabled for the specified port.	**# show port header-strip port-list 1/1/c1**
Displays the list of ports on which the Header Stripping protocol is enabled for all the box IDs.	**# show port header-strip box-id all**
Displays the list of ports on which the specified Header Stripping protocol is enabled for all the box IDs.	**# show port header-strip box-id all type <mpls-l3|vxlan| mpls-advanced>**
Displays the VLAN resources that are available.	**#show vlan-resource**
Deletes the utilization alarm threshold for a specified port.	**(config) # no port 1/1/x1 alarm buffer-threshold**
Deletes the utilization alarm threshold for a specified port in a specified direction.	**(config) # no port 1/1/x1 alarm buffer-threshold rx**
Deletes the port alias for a specified port.	**(config) # no port 1/1/x1 alias**
Deletes all assigned roles.	**(config) # no port 1/1/x1 assign all**
Deletes a specified assigned role for a specified port.	**(config) # no port 1/1/x1 assign role admin**
Deletes a comment for a specified port.	**(config) # no port 1/1/x1 comment**
Disables outer VLAN stripping on specified egress ports.	**(config) # no port 1/2/x1 egress-vlan strip**
Deletes VLAN tagging for a specified port.	**(config) # no port 1/1/x1 ingress-vlan-tag**

Task	Command
Unlocks a specified port.	**(config) # no port 1/1/x1 lock**
Does not allow any user to share lock privileges.	**(config) # no port 1/1/x1 lock-share all**
Does not allow a specified user to share lock privilege.	**(config) # no port 1/1/x1 lock-share user operator**
Clears the port breakout mode.	**(config) # no port 1/1/c1 mode** or **(config) # port 1/1/c1 mode none**
Deletes all timestamp bytes appended to a specified port in the ingress direction.	**(config) # no port 1/1/x1 timestamp append-ingress**
Deletes all timestamp bytes stripped from a specified port in the egress direction.	**(config) # no port 1/1/x1 timestamp strip-egress**
Deletes all roles from a specified port's share-tool-mirror list.	**(config) # no port 1/1/x1 tool-share all**
Deletes a specified role from a specified port's share-tool-mirror list.	**(config) # no port 1/1/x1 tool-share role monitor**
Disables the PTP configurations on a particular port. However, the PTP VLAN ID for the port will be displayed in the "**show running config**" output.	**(config) # no port 1/1/c1 ptp enable**
Removes the L2GRE ID configured for the specified network port.	**(config) # no port 1/1/x1 l2gre-id**
Removes the VXLAN ID configured for the specified network port.	**(config) # no port 1/1/x1 vxlan-id**
Displays the ingress or egress timestamp details in a table format.	**(config) #show timestamp brief**
Disables the Header Stripping protocol on the specified port.	**(config) # no port 1/3/x16 header-strip**
Disables the Header Stripping protocol on all the ports.	**(config) # no port all header-strip**
Clears the port filter statistics for all the line cards.	**(config) # clear port filter stats all**
Clears the port filter statistics for a line card placed inside the specified box.	**(config) # clear port filter stats box-id 2**
Clears the port filter statistics for the specified list of ports.	**(config) # clear port filter stats port-list 2/1/x1..x4**
Clears the port filter statistics for a line card placed at the specified slot.	**(config) # clear port filter stats slot 1**
Displays configuration, part numbers, App code, grid frequency, grid spacing, and Transmit (Tx) optics power for each port	**(config) # show port zr-optics**

port-group

Required Command-Line Mode = Configure

Use the **port-group** command to create groups of network or tool ports. Ports can belong to multiple groups. However, you cannot mix port types in a single group, and the ports within a port group must be on the same chassis.

Starting in software version 4.8, port groups used in GTP overlapping maps support GigaStream.

Starting in software version 5.1, port groups support a list of tunnel endpoints.

Port groups are used to simplify administration of Deep Observability Pipeline ports, allowing you to group ports with a similar purpose for convenience in identification.

NOTE: Keep in mind the following points when using port groups together with role-based access:

- To access **any** port in a port group, a user must have roles assigned that grant access to **all** ports in the port group.
- If a user has different permissions on different ports in a port group, the system will assign the user the lowest of those assigned permissions for all ports in the port group.

The **port-group** command has the following syntax:

```
port-group alias <alias>
  comment <comment>
  gigastream-list <list of GigaStream aliases>
  port-list <port-id | port-alias | port-list | inline-network-alias | inline-network-group-alias>
  smart-lb <disable | enable>
  te-list <list of tunnel endpoints> or <range of tunnel endpoints>
  weight <port ID | te-id> <1-100>
```

The following table describes the arguments for the **port-group** command:

Argument	Description
alias <alias>	Specifies a name for the port group. For example: **(config) # port-group alias pg1**
comment <comment>	Optionally, you can supply a comment for the port group. For example: **(config) # port-group alias pg1 port-list 2/1/x1..x2 comment “Port Group 1”**

Argument	Description
gigastream-list <list of GigaStream aliases>	Specifies a list of GigaStreams to include in this port group. This parameter is only supported for GTP forward listing and GTP flow sampling, in which port groups are used in overlapping maps. For example: **(config) # port-group alias pg1 gigastream-list GTP-sample1** Refer to the *"GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide*.
port-list <port-id | port-alias | port-list | **inline-network-alias |** **inline-network-group-alias>**	Specifies the ports to include in this port group. Use one of the following: • **port-id, port-alias, port-list**—Specifies a port using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **inline-network-alias**—Specifies an inline network alias. • **inline-network-group-alias**—Specifies an inline network group alias.
smart-lb <disable | enable>	Enables or disables GigaSMART load balancing. The default is disabled. For example: **(config) # port-group alias pg1 smart-lb enable** To use the **te-list** parameter of the **port-group** command for tunnel load balancing for L2GRE tunnel encapsulation, **smart-lb** must be enabled.
<list of tunnel endpoints> or <range of tunnel endpoints>	Specifies the list of destinations to which to send from the tunnel to the tunnel endpoints. This parameter is only supported for Layer 2 GRE encapsulation. Specify the tunnel endpoint identifiers or aliases, or a range of tunnel endpoint identifiers using the following syntax: • Tunnel endpoint identifiers and/or aliases in a comma separated list. The maximum number supported in the list is 16. • Tunnel endpoint identifiers as a range. Only tunnel endpoint identifiers are supported in the range. Examples: **(config) # port-group alias pg1 te-list te1,te2,te3** **(config) # port-group alias pg1 te-list teAlias1,teAlias2** **(config) # port-group alias pg1 te-list teAlias1,te2** **(config) # port-group alias pg1 te-list te1..te20** **NOTE:** A mix of tunnel endpoints with **port-list** or **gigastream-list** is not supported in a port group.
weight <port ID | te-id> <1-100>	Specifies load balancing weights for the ports in the port list or the tunnel endpoints in the tunnel endpoint list. Weights apply to Weighted Round Robin (wt-round-robin), Weighted Least Bandwidth (wt-lt-bw), Weighted Least Packet Rate (wt-lt-pkt-rate), Weighted Least Connection (wt-lt-conn), and Weighted Least Cumulative Traffic (wt-lt-tt-traffic) load balancing metrics. Weighted Least Bandwidth (wt-lt-bw) is not support for tunnel.

Argument	Description
	The value of weight is from 1 to 100. The default is 1. Weight is optional for tunnel endpoints. Weight is specified on one tunnel endpoint at a time. Examples: **(config) # port-group alias pg2 weight 2/1/x1 30** **(config) # port-group alias pg2 weight te1 50**

Related Commands

The following table summarizes other commands related to the **port-group** command:

Task	Command
Displays port groups.	**# show port-group**
Displays detailed information for a specified port group.	**# show port-group alias pg1**
Displays all port groups.	**# show port-group all**
Displays all port groups in table format.	**# show port-group brief**
Displays load balancing statistics for a specified port group.	**# show load-balance port-group stats alias portgrp1**
Displays load balancing statistics for all port groups. **NOTE:** • After the delete request and response, the **Active Sessions** count stats in the output of **"show load-balance port-group stats all"** command will be cleared only after 4 minutes approximately. • In Standalone UPN, the LB **Active Sessions** count may display incorrect values in case of engine grouping.	**# show load-balance port-group stats all**
Deletes a specified port group.	**(config) # no port-group alias pg1**
Deletes a GigaStream list from a specified port group.	**(config) # no port-group alias pg1 gigastream-list**
Deletes a port list from a specified port group.	**(config) # no port-group alias pg1 port-list**
Deletes a tunnel endpoint list from a specified port group.	**(config) # no port-group alias pg1 te-list**

Task	Command
Deletes all port groups.	**(config) # no port-group all**

port-pair

Required Command-Line Mode = Configure

Use the **port-pair** command to configure a pair of network ports within the same GigaVUE-OS® HC Series node. A port pair refers to a bidirectional connection where all traffic (excluding Ethernet control frames which are discarded in hardware) arriving at one port in the pair is transmitted out of the other port, and vice versa, functioning as a passthrough TAP.

Notes on Port-Pairs

- Port-pairs cannot be established between any other ports on the same TAP-HC0-G100C0 or TAP-HC1-G10040 and another TAP-HC0-G100C0 or TAP-HC1-G10040 on the same chassis. Port-pairs can only be established between the same TAP pairs.
- Port-pairs support link status propagation – when one port goes down, the other port goes down (and vice-versa).

NOTE: A port-pair created on a copper TAP has LFP enabled by default.

The **port-pair** command has the following syntax:

```
port-pair alias <alias>
   between <<port ID> | <port alias> and <port ID> | <port alias>>
     [comment <comment>]
    [lfp <enable | disable>]
```

The following table describes the arguments for the **port-pair** command:

Argument	Description
alias <alias>	Specifies a name for the port pair.
between <<port ID | port alias> and <port ID | port alias>> [comment <comment>]	Specifies the ports in the port pair using a port ID or port alias. A port ID is the numerical identifier of a port in box ID/slot ID/port ID format (for example, 1/1/x1). You can also identify either of the ports in the port pair by alias, if configured. Port aliases are configured using the port alias command. Refer to port for more information. You can supply an optional comment for the port pair. For example: **(config) # port-pair alias AtoB between portA and portB comment "from A to B"**
lfp <enable |	Specifies link failure propagation (LFP). Port pairs can operate with or without LFP as

Argument	Description
disable>	follows: • With LFP enabled, link failure on one of the ports in the port pair automatically brings down the opposite side of the port pair. • With LFP disabled, the opposite port is not brought down automatically. **NOTE:** A port pair created on a copper TAP has LFP enabled by default. For example: **(config) # port-pair alias AtoB lfp disable**

Related Commands

The following table summarizes other commands related to the **port-pair** command:

Task	Command
Displays all port pairs.	**# show port-pair**
Displays information for a specified port pair.	**# show port-pair alias AtoB**
Displays all port pairs.	**# show port-pair all**
Displays all port pairs in table format.	**# show port-pair brief**
Deletes all port pairs.	**(config) # no port-pair all**
Deletes a specified port pair.	**(config) # no port-pair alias AtoB**

ptp

Required Command-Line Mode = Configure

Required User Level = Admin

Refer to the following sections for information about how to use the **ptp** command on the various Gigamon devices:

- Use ptp on GigaVUE-TA200 Devices

Use ptp on GigaVUE-TA200 Devices

Use the **ptp** command to configure the PTP domain, clock mode, and priority on the GigaVUE-TA200 devices. Based on these PTP parameters, the primary source will be selected from the clocks in a network.

Refer to the *"Timestamps"* chapter in the *GigaVUE Fabric Management Guide* for more information about how to use PTP to timestamp packets.

Use the following syntax to configure PTP globally for a device:

```
ptp
  alias <string>
  domain <value>// Range is 0, 24–43
  mode <ordinary | boundary>
  local-priority <value> // Range is 1—255
  priority2 <value> // Range is 0—255
```

Use the following syntax to configure PTP globally for a cluster:

```
ptp
  box-id <box-id/all> alias <string>
  domain <value>// Range is 0, 24–43
  mode <ordinary | boundary>
  local-priority <value> // Range is 1—255
  priority2 <value> // Range is 0—255
```

The following table describes the arguments for the **ptp** command:

Argument	Description
alias <string>	Specifies the name of the PTP for a device. For example: **(config) # ptp alias <string>**
box-id <box-id/all> alias <string>	Specifies the name of the PTP for all devices in a cluster. Ensure that you provide a unique name for each device in the cluster. For example: **(config) # ptp box-id <box-id/all> alias <string>**
domain <value>	Specifies a PTP domain number that will be used for the clock. The valid range includes 0 and the numbers between 24 and 43. • Domain 0 for IEEE 1588 default profile. • Domain 24 - 43 for ITU-T G.8275.1 Telecom profile. For example: **(config ptp alias <string>) # domain 24** You can also choose to configure two attributes in a single line. For example, you can configure the PTP alias and domain in a single line as follows: **(config) # ptp alias <string> domain 24**
mode <ordinary | boundary>	Configures the mode for PTP, ordinary or boundary. For example: **(config ptp alias <string>) # mode ordinary**
local-priority <value>	Specifies a priority value that will be used to determine the PTP source when the BMC algorithm chooses more than one clock as the source. This value overrides the default criteria for the BMC selection. The clock with the lower priority value will be selected. For example, on a specific device, the local priority of a port is compared with the local priority of the clock. Based on the following criteria, the source is determined:

Argument	Description
	• If the local priority of the port is greater than the local priority of the clock, the clock becomes the source. • If the local priority of the port is lesser than the local priority of the clock, the pSort becomes the source. • If the local priority of the port is equal to the local priority of the clock, the clock identity of the port and the clock is compared and the one that has the lower value becomes the source. The valid range is between 1 and 255. The default is 128. For example: **(config ptp alias <string>) # local-priority 128**
priority2 <value>	Specifies a priority value that will be used to determine the primary source in a network. For example, if there are two clocks in a network that match the default criteria, the clock that has the lower priority value will be selected as the primary source. The valid range is between 0 and 255. The default is 128. For example: **(config ptp alias <string>) # priority2 128**

Related Commands

The following table summarizes other commands related to the **ptp** command:

Task	Command
Displays PTP runtime state.	**(config) # show ptp alias <string>**
Displays PTP runtime state for a cluster.	**(config) # show ptp box-id <box-id/all> alias <string>**
Displays PTP configuration.	**(config) # show ptp alias <string> configured**
Displays PTP configurations in table format.	**(config) # show ptp alias <string> brief**
Displays the PTP clock details.	**(config) # show ptp alias <string> clock**
Displays the PTP's foreign source record for a standalone node and a cluster.	**(config) # show ptp port-list <x/y/z> foreign-source**
Displays the PTP's parent clock information.	**(config) # show ptp alias <string> parent**
Displays the PTP statistical counters.	**(config) # show ptp alias <string> counters**
Displays PTP configurations for a cluster.	**(config) # show ptp box-id <box-id/all> alias <string> configured**
Displays PTP configurations for a cluster in table format.	**(config) # show ptp box-id <box-id/all> alias <string> brief**

Task	Command
Displays the PTP clock details for a cluster.	**(config) # show ptp box-id <box-id/all> alias <string> clock**
Displays the PTP's parent clock information for a cluster.	**(config) # show ptp box-id <box-id/all> alias <string> parent**
Displays the PTP statistical counters for a cluster.	**(config) # show ptp box-id <box-id/all> alias <string> counters**
Displays the list of ports on which PTP is enabled.	**(config) # show ptp port-list <port-list>**
Displays the PTP statistical counters for a list of PTP-enabled ports.	**(config) # show ptp port-list <port-list> counter**
Clears the statistical counters for the specified PTP.	**(config) # clear ptp alias <string> counters**
Clears the PTP statistical counters for the specified cluster.	**(config) # clear ptp box-id <box-id/all> alias <string> counters**
Clears the PTP statistical counters for a list of PTP-enabled ports.	**(config) # clear ptp port-list <port-list> counter**

radius-server

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **radius-server** command to specify the RADIUS servers to be used for authentication. You can specify multiple RADIUS servers. Servers are used as fallbacks in the same order they are specified—if the first server is unreachable, the second is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

Refer to Add a RADIUS Server for examples of adding and configuring a RADIUS server.

The **radius-server** command has the following syntax:

```
radius-server
   extra-user-params roles enable
  host <IPv4/IPv6 address or hostname> [auth-port <port-number>] [enable] [shared-secret
<string> | 9 <encrypted string>] [prompt-secret <string>] ]      [retransmit <retries>] | [timeout
<seconds>]
  shared-secret <string> | 9 <encrypted string>
  retransmit <retries>
  timeout <seconds>
```

The following table describes the arguments for the **radius-server** command. The **shared-secret**, **retransmit**, and **timeout** values can be specified both globally and on a per-host basis. Per-host values override any configured global values.

Argument	Description
extra-user-params roles enable	Enables theGigaVUE HC Series node to accept user roles assigned in the RADIUS server. Note that the role name must match a role configured on the local node or cluster. Refer to the Configure AAA for details. For example: **(config) # radius-server extra-user-params roles enable**
host <IPv4/IPv6 address or hostname>	Specifies the IP address (IPv4 or IPv6) or hostname of the RADIUS server. The same IP address can be used for more than one RADIUS server so long as they use different **auth-port** values. Examples: **(config) # radius-server host 1.1.1.1 (config) # radius-server host 2001:db8:a0b:12f0::11 key gigamon enableconfig) # radius-server host www.MyCo.com**
auth-port <port-number>	Specifies the UDP port number on which the RADIUS server is running. If included, the **auth-port** must be specified immediately after the host IP address. If you do not specify a port, the default RADIUS authentication port number of 1812 is used. For example: **(config) # radius-server host 1.1.1.1 auth-port 123**
enable	Administratively enables this RADIUS server. For example: **(config) # radius-server host 1.1.1.1 auth-port 123 enable**
shared-secret <string> | 9 <encrypted string>	Specifies the shared secret key to be used for encryption of authentication packets sent between the GigaVUE HC Series node and this specific RADIUS server. Any value specified here will override the shared secret key specified in the **radius-server host** command. For example: **(config) # radius-server host 1.1.1.1 auth-port 123 enable shared-secret admin12** To configure an encrypted shared-secret, use 9 keyword. **(config) # radius-server host 1.1.1.1 auth-port 123 enable shared-secret 9 8IEiQIIyQ1JhfkfhFBgAds/yjxHIhvTCkEakIWJivnA=**
prompt-secret	Requires the user to enter the shared secret string during login. This option is mutually exclusive with the **key** option. For example: **(config) # radius-server host 1.1.1.1 auth-port 123 enable shared-secret admin12 prompt-secret**
retransmit <retries>	Specifies the number of times the GigaVUE HC Series node will attempt to authenticate with this specific RADIUS server. Any value specified here will override the global value specified in the **radius-server retransmit** command. The valid range is from 0 to 5. The default is 1. To disable retransmissions, use 0. For example: **(config) # radius-server host 1.1.1.1 auth-port 123 enable retransmit 4**
timeout <seconds>	Specifies how long theGigaVUE HC Series node should wait for a response from this specific RADIUS server to an authentication request before declaring a timeout failure. Any value specified here will override the global value specified in the **radius-server timeout**

Argument	Description
	command. The valid range is from 0 to 60 seconds. The default is 5 seconds. For example: **(config) # radius-server host 1.1.1.1 auth-port 123 enable timeout 20**
shared-secret <string> | 9 <encrypted string>	Specifies a global shared secret string to be used for encryption of authentication packets sent between theGigaVUE HC Series node and all RADIUS servers. The global value can be overridden with the shared secret specified in the **radius-server host** command. For example: **(config) # radius-server shared-secret admin12** To configure an encrypted shared-secret, use 9 keyword. **(config) # radius-server shared-secret 9 8IEiQIIyQ1JhfkfhFBgAds/yjxHIhvTCkEakIWJivnA=**
retransmit <retries>	Specifies a global vGigaVUE HC SeriesGigaVUE-OS HC Series node will attempt to authenticate with a RADIUS server. The global value can be overridden with the retransmit value specified in the **radius-server host** command. The valid range is from 0 to 5. The default is 2. To disable retransmissions, use 0. For example: **(config) # radius-server retransmit 3**
timeout	Specifies a global value for how long the GigaVUE HC Series node should wait for a response from the RADIUS server to an authentication request before declaring a timeout failure. The global value can be overridden with the timeout value specified in the **radius-server host** command. The valid range is from 0 to 60 seconds. The default is 5 seconds. For example: **(config) # radius-server timeout 20**

Related Commands

The following table summarizes other commands related to the **radius-server** command:

Task	Command
Displays the list of configured RADIUS servers and related RADIUS settings.	**# show radius**
Does not allow the RADIUS server to include additional roles for a remotely authenticated user in the response.	**(config) # no radius-server extra-user-params roles enable**
Deletes a RADIUS host with the specified IPv4 or IPv6 address, or hostname.	**(config) # no radius-server host 1.1.1.1** **(config) # no radius-server host www.MyCo.com**
Deletes a RADIUS host on a specified port.	**(config) # no radius-server host 1.1.1.1 auth-port 234**
Administratively disables the specified RADIUS server on the specified port.	**(config) # no radius-server host 1.1.1.1 auth-port 234 enable**

Task	Command
Administratively disables the specified RADIUS server on the default port.	**(config) # no radius-server host 1.1.1.1 enable**
Deletes the global RADIUS server shared secret key.	**(config) # no radius-server shared-secret**
Resets the global RADIUS server retransmit count to the default.	**(config) # no radius-server retransmit**
Resets the global RADIUS server timeout settings to the default.	**(config) # no radius-server timeout**

redundancy-profile

Required User Level = Admin

Use the **redundancy-profile** command to configure an inline redundancy profile.

This command is only applied to GigaVUE-HC3, and GigaVUE-HC1 nodes.

Refer to inline-network to configure the alias of the redundancy profile, which must be configured prior to specifying **redundancy-profile** arguments.

Refer to the "*Configure Gigamon Resiliency for Inline Protection*" section in the *GigaVUE Fabric Management Guide* for details.

The **redundancy-profile** command has the following syntax:

```
redundancy-profile alias <alias>
  protection-role <primary | secondary | suspended>
  signaling-port <port ID or port alias>
```

The following table describes the arguments for the **redundancy-profile** command:

Argument	Description
alias <alias>	Specifies the name of the redundancy profile. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # redundancy-profile alias RP1**
protection-role <primary | secondary | suspended>	Specifies the protection role of a redundancy profile for the inline network as follows: • **primary**—Specifies the primary (active) protection role. • **secondary**—Specifies the secondary (standby) protection role. • **suspended**—Specifies that the protection role is on hold. When suspended, the protection role is on hold. Changing a GigaVUE-OS node from the primary role to the suspended role can be used to manually force the secondary node to become active. The suspended role is also used when performing maintenance.

Argument	Description
	The default is **suspended**. For example: **(config) # redundancy-profile alias RP1 protection-role primary**
signaling-port <port ID or port alias>	Specifies the signaling port to be used to connect two GigaVUE HC Series nodes. Either a port ID or a port alias can be specified. Examples: **(config) # redundancy-profile alias RP1 signaling-port 1/1/x1** **(config) # redundancy-profile alias RP1 signaling-port inNet**

Related Commands

The following table summarizes other commands related to the **redundancy-profile** command:

Task	Command
Displays redundancy profile and redundancy control state for an inline network.	**# show inline-network**
Displays all redundancy profiles.	**# show redundancy-profile**
Displays a specified redundancy profile.	**# show redundancy-profile alias new**
Displays all redundancy profiles.	**# show redundancy-profile all**
Deletes a specified redundancy profile.	**(config) # no redundancy-profile alias new**
Deletes all redundancy profiles.	**(config) # no redundancy-profile all**

reload (reboot)

Required Command-Line Mode = Enable

Required User Level = Admin

Use the **reload** command to reboot or halt the system.

NOTE: Rebooting control card 1 also reboots control card 2, interrupting any ongoing sessions there. Active sessions with control card 2 are not common and will typically only occur when updating the software on the card.

The **reload** command has the following syntax:

```
reload
  force [immediate]
  halt
```

Use the **reload** command without any arguments to reboot the system.

The following table describes the arguments for the **reload** command:

Argument	Description
force [immediate]	Reboots the system immediately, regardless of whether the system is currently busy. Optionally, use **immediate** to ensure an immediate reboot. For example: **(config) # reload** **(config) # reload force** **(config) # reload force immediate**
halt	Stops all system activities without powering the system down. For example: **(config) # reload halt**

reset

Required Command-Line Mode = Configure

Required User Level = Admin

The **reset** command returns the GigaVUE-OS® HC Series and GigaVUE-OS® TA Series nodes to factory settings.

The **reset** command has the following syntax:

```
reset factory <all | all-with-halt | keep-all-config | keep-only-syslog | keep-ports | only-traffic>
```

The system reboots or halts after the process is completed.

The following table describes the arguments for the **reset** command:

Command	Description
reset factory <all | all-with-halt | keep-all-config | keep-only-syslog | keep-ports | only-traffic>	Specifies the portions of the system configuration to reset to factory defaults: • **all**— Resets all configuration files and log files, then reboots the system • **all-with-halt**—Resets all configuration files and log files, then halts the system instead of rebooting. • **keep-all-config**—Retains all configuration files but clears all log files, then reboots the system. Note that this command does not affect any

Command	Description
	configuration settings. • **keep-only-syslog**—Retains all log files but resets all configuration files, then reboots the system. • **keep-ports**— Retains system, chassis, cards, ports configuration and log files, but resets all other configuration files, then reboots the system. • **only-traffic**— Resets chassis, cards, ports, and traffic configuration, but retains system configuration, then reboots the system. Examples: **(config) # reset factory all** **(config) # reset factory all-with-halt** **(config) # reset factory keep-all-config** **(config) # reset factory keep-only-syslog** **(config) # reset factory keep-ports** **(config) # reset factory only-traffic** **NOTE:** The commands **reset factory all**, **reset factory all-with-halt** and **reset factory keep-only-syslog** delete passwords on user accounts. When user Log in with the **admin** account, user will be prompted for a new password. **NOTE:** The commands **reset factory all**, **reset factory all-with-halt**, and **reset factory keep-only-syslog** enables ZTP by default, prompting it to automatically obtain an IP address and configuration boot file. **NOTE:** The commands **reset factory keep-all-config**, **reset factory Keep-only-syslog**, **reset factory keep-ports** and **reset factory only-traffic** applies to all node in the cluster.

serial

Required Command-Line Mode = Standard

Use the **serial** command to configure the serial console port.

The **serial** command has the following syntax:

```
serial
  baudrate <9600 | 115200>
  enable
```

The following table describes the arguments for the **serial** command:

Argument	Description
baudrate <9600 | 115200>	Specifies the baud rate for the serial console port. The valid values are 9600 or 115200. The default is 115200. For example: **(config) # serial baudrate 9600** **NOTE:** The following products support only the serial baud rate of 115200: • GigaVUE-HC3 • GigaVUE-HC1 • GigaVUE-TA100 • GigaVUE-TA200 • GigaVUE-TA25
enable	Enables the console port. For example: **(config) # serial enable**

Note the following about this CLI command:

- To apply changes to the console port's speed, save the active configuration and reload the system.
- After setting the console port to 9600 or 115,200 bps using the **serial baudrate** command, the bootloader output will appear correctly on the connected serial console during a system boot provided the client application is set to run at a matching speed.
- When the serial baudrate is set to 9600 bps, you will need to press the spacebar twice for the CLI login prompt to appear following a reboot.

Related Commands

The following table summarizes other commands related to the **serial** command:

Task	Command
Displays serial console settings.	**# show serial**
Disables serial console access.	**(config) # no serial enable**

sfp

Required Command-Line Mode = Configure

The **sfp** command is reserved for future use.

sffp profile

Required Command-Line Mode = Configure

Use the **sffp profile** command to configure the Control and User Plane Separation (CUPS) profile on a Control Processing Plane (CPN) for routing the Transport Agent (GTA) packets.

The **sffp profile** command has the following syntax:

```
gpfcp-profile alias <alias>
 profile <add | delete>
add
  ip interface <interface>
  port-list <port number>
    type <control | user>
    ips <interface>
```

The following table describes the arguments for the **sffp profile**command:

Argument	Description
sffp-profile alias <alias>	Specifies the name of the sffp profile. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # sffp-profile alias sffp-profile-1**
profile <add | delete>	Specifies to add a sffp profile or delete a sffp profile.
add ip interface <interface> port-list <number> type <control | user> sx-ips <interface>>	Specifies to add the attributes of a sffp profile. The attributes are as follows: • **ip interface <interface>** - Specifies the IP address of the IP interface. • **port list <port number>**- Specifies the port number. The value ranges from 1 to 65535 • **type** Specifies any one of the following node type: o **control** - Specifies the control node type for CPN. You can specify only a single port as control node type. o **user** - Specifies the following user interface node type for UPN. You can specify multiple ports as user interface node type . o **sx-ips <interface>**- In 4G CUPS, specifies the IP addresses of S11 SGW-C , S1 SGW-U, S5c PGW-C and S5u PGW-U interfaces of SFFP profile. o **sx-ips <interface>**- In 5G CUPS, specifies the IP addresses of N3 SMF and N11 UPF interfaces of SFFP profile. For example: **(config) # profile add ip-interface 11.0.0.2 port-list 5000 type user sx-ips 10.10.6.7,10.10.6.8** **Note**: The system supports 128 IPv6 or 256 IPv4 addresses for SFFP profile.

Related Commands

The following table summarizes the command related to the **sffp-profile**command:

Task	Command
Displays a specified sffp profile	**show sffp-profile alias sffp-profile-1**

show

Required Command-Line Mode = Standard

Use the **show** command to view configuration and status information for GigaVUE-OS nodes. Configuration commands in the CLI have corresponding **show** commands that display the configured values and the current status.

To match a keyword or a regular expression in the **show** command, use the option **matching <regex>** after the **show** command. You should type the keyword completely as the auto-fill option is not supported for the keyword. The command has the following syntax;

show <command> matching <regex>

<regex> is the regular expression to be used as a filter.

For example:

```
Example 1
show port params matching SFP

Example 2
show chassis matching down

Example 3
show running-config matching apps
```

NOTE: For the following three commands, matching option is not available:
show cli history matching
show diag matching
show system matching

The device does not cancel an operation when <CTRL+C> is given after a show command instead the Command Line Interface shows an error message. In stand-alone devices, no other command should be issued after the <CTRL+C> command. You can use either of the workaround:

- Reboot the device from the GigaVUE-FM
- Exit from the CLI session and then re-log

The following table describes the **show** commands:

Show Command	Description
show aaa **show aaa authentication attempts** **show aaa authentication attempts configured** **show aaa authentication attempts status [user <username>]** **show aaa authentication certificate crl name default**	For usage examples, refer to aaa accounting, aaa authentication, and aaa authorization.
show action [alias <alias> | detail]	For usage examples, refer to policy.
box <box ID> [brief] | brief | slot <slot ID> [brief]]	For usage examples, refer to filter-template.
show apps	
show apps <asf | enhanced -asf | enhanced- lb | enhanced-slicing | exporter | gtp-whitelist | hsm | hsm-group |hsm-group luna-key labels| inline-ssl | keystore | listener | metadata | netflow | port-throttle | regex-profile | sip-whitelist | split-dms | ssl>	
show apps asf <alias <alias> | all | stats> <alias <alias> | all>	For usage examples, refer to apps asf
show apps enhanced-asf <alias <alias> | all | stats> <alias <alias> | all> **show apps enhanced-lb <alias <alias> | all>>**	For usage examples, refer to apps enhanced asf..
show apps enhanced-slicing <alias <alias> | all | stats>	For usage examples, refer to apps enhanced-slicing.
show apps exporter alias <alias>	For usage examples, refer to apps exporter.
show apps exporter stats all	For usage examples, refer to apps exporter.
show apps exporter tls-pcapng session any	
show apps gtp-whitelist alias <alias> count	For usage examples, refer to apps gtp-whitelist.
show apps hsm <alias <alias> | all>>	For usage examples, refer to apps hsm.
show apps hsm-group <anonkneti | enquiry |	

Show Command	Description
chkserv | ckinfo | key | world | config | module | session-stats | buffer-stats | all | status>	For usage examples, refer to apps hsm-group.
show apps inline-ssl **caching** **certificate validation <certificate CN | status>** **url <domain name | status>** **global** **monitor <session <any | match <ipv4-dst <IP address | mask> | ipv4-src | <IP address | mask>** **<ipv6-dst <IPv6 address | mask> | ipv6-src |** **<IPv6 address | mask>l4port-dst <L4port | any> | l4port-src <L4port |any>>> | summary>** **profile <alias <alias> [decryptlist | nodecryptlist] <domain name> | all>** **session** **any** **debug vport <vport alias>** **match ipv4-src <IP addr/mask> ipv4-dst <IP addr/** **mask> ipv6-src <IPv6 addr/mask> ipv6-dst** **<IPv6 addr/mask> l4port-src <port number> l4port-dst <portnumber> [detail] | hostname <hostname>** **summary** **trust-store <all | certificate fingerprint <fingerprint string>>**	For usage examples, refer to apps inline-ssl. Debug is reserved for internal use. **NOTE:** IPV6 traffic decryption is supported only for GEN 3 cards. Refer to the GigaVUE-HC1 Hardware Installation Guide and GigaVUE-HC3 Hardware Installation Guide for the list of GEN 3 card numbers.
show apps keystore <alias <alias> <certificate | summary>| all>	For usage examples, refer to apps keystore.
show apps listener alias <alias>	For usage examples, refer to apps listener.
show apps listener tls-pcapng session any	

Show Command	Description
show apps metadata **app-profile [alias <alias> | all]** **cache [alias <alias> | all]** **exporter [alias <alias> | all]** **monitoring [all | port-list <alias>]** **stats [cache [alias <alias> | all]** **exporter [alias <alias> | all]** **mgmt interface [all | exporter alias <alias>]** **monitoring [all | port-list <alias>]**	
show apps netflow [exporter | monitor | port-id | record] [alias <alias> | all | stats [alias <alias> | all]]	
show apps port-throttle <alias <alias> | all>	For usage examples, refer to apps netflow.
show apps regrex-profile <alias <alias> | all>	For usage examples, refer to apps regrex-profile.
show apps sip-whitelist <alias <alias> | all>	For usage examples, refer to apps sip-whitelist.
show apps split-dns profile <alias <alias> | all>	For usage examples, refer to apps split-dns
show apps ssl **key <alias <alias> | all>** **service <alias <alias> | all> stats [alias <alias> | all]**	For usage examples, refer to apps ssl.
show crypto acme client info	Displays the acme client information and the status of the certificate. For usage examples, refer to crypto.
show banner	For usage examples, refer to banner.
show battery	For usage examples, refer to battery.
show battery optimization	For usage examples, refer to battery optimization.
show bonds [bonded interface]	For usage examples, refer

Show Command	Description
	to bond.
show buffer **box-id <box ID>** **port <port-list> [rx | tx]** **slot <slot ID>**	For usage examples, refer to card (GigaVUE-OS® HC Series) and port.
show buffer-index	For usage examples, refer to port.
show cards [box-id <box ID> | slot <slot ID>]	For usage examples, refer to card (GigaVUE-OS® HC Series).
show chassis [box-id <box ID>] [faceplate-numbering]	Displays ONIE faceplate numbering to GigaVUE-OS faceplate numbering. This command only applies on a white box. For more information, refer to White Box Port and Faceplate Labeling.
show cli [history [number of lines]]	For usage examples, refer to cli.
show clock	For usage examples, refer to clock.
show cluster **box-id <box ID>** **configured** **global [brief]** **history [box-id <box ID>]** **local [error-status]** **leader** **node <node ID>** **standby**	For usage examples, refer to cluster.
show condition [alias <alias> | detail]	For usage examples, refer to policy.
show configuration **audit** **files [filename | initial]** **full** **running [full]** **text files**	For usage examples, refer to configuration.

Show Command	Description
show crypto certificate **ca-list [default-ca-list]** **default-cert [detail | public-pem]** **detail** **name <cert-name> [system-self-signed] [detail | public-pem]** **public-pem**	For usage examples, refer to crypto.
show diag **show diag [detail]** **show diag detail upload <upload URL>**	Displays diagnostics for trouble-shooting. Displays diagnostic information about fabric statistics, system-health, and inline-ssl statistics detail, in addition to the diagnostic information displayed in **show diag**. Uploads the output of **show diag detail** to the specified URL.
show egress-vlantag	For usage examples, refer to port.
show email [dead-letter | events]	For usage examples, refer to email.
show environment **box-id <box ID> slot <slot ID> type <fan | temperature | voltage | psu>**	Displays environment information.
show environment type psu psu-detail [all | psu-id]	Displays the PSU diagnostic attributes parameters for all the PSU modules or for the specified PSU module id (**psu-id**). Depending on the number of PSU modules in the chassis, **psu-id** can be 1, 2, etc. The nomenclature for the power module parameter is as follows: • N/A - Not Applicable • OK - Fault has not set • Check - Fault has set • Read Err - Could not read

Show Command	Description
	Rules and Notes Keep in mind the following exceptions and limitations: • If a GigaVUE node is unable to read the product code of the Power Supply Unit (PSU) module, then all the PSU parameters will be displayed as N/A except the following parameters: o Product Code o Serial Number o Hardware Revision o Status • In addition to the above parameters, for GigaVUE-HC3, the value for the "capacity" parameter will also be displayed as N/A. • Minimal output voltage (such as 0.63 V) and minimal output current (such as 0.2 A) is expected when the PSU module is in the "off" state. This behavior is due to a voltage leak. In GigaVUE-HC3 node, the input under "voltage fault" is set only when there is a real under-voltage condition. • In some modules, the temperature fault is latched, which means that if an over-temperature condition occurs, the temperature fault is set. When the PSU temperature drops and remains within a safe range, the PSU module will shut down

Show Command	Description
	and will restore power automatically, but the temperature fault will remain set. The only time the fault setting is cleared is when the PSU is power-cycled. This behavior occurs in the following modules: o GigaVUE-TA200 o GigaVUE-HC1 • When using HC1-Plus with certain batches of PSU modules, if you issue the CLI command "show environment type psu psu-detail" while one of the power modules is powered off (for example, by removing the power cable), the powered-off module may still display non-zero values for "Input Voltage," "Output Voltage," "Input Current," and "Output Current." The power module firmware reports the wrong value when it is in the powered-off state. You can ignore this issue if it occurs, as it does not affect the functionality of the PSU module.
show files <debug-dump | pcap | system [detail] | tcpdump>	For usage examples, refer to file and pcap.
show filter-resource [all] | [box-id <box ID>] | [slot-id <slot ID>] [brief]	For usage examples, refer to filter-template.
show filter-template [alias <alias> [brief] | all | limit [all | box <box ID> | slot <slot ID>]	For usage examples, refer to filter-template.
slot ID > [brief]]	For usage examples, refer to chassis.

Show Command	Description
engine <port-list> <arp | details | stats> | interface [eth2 | eth3] <vlan <VLAN ID>> <arp | details | stats>	For usage examples, refer to gigasmart.
show gigastream **advanced-hash [box-id <box ID> | brief | slot <slot ID>** **alias <alias>** **all** **brief**	For usage examples, refer to gigastream.
show gsgroup **alias <alias>** **all** **flow-ops-report alias <alias> type flow-sampling | ssl-decryption <any | device-ip-mask <IP address> <netmask> |flow-filtering <gtp-imsi-pattern | imei-pattern | msisdn-pattern > [summary | upload <upload URL>] flow-sip <any |** **callerid-pattern> [summary | upload <upload URL>]** **inline-ssl any upload <upload URL>** **flow-5g <supi-pattern | pei-pattern | dnn-pattern > [summary | upload <upload URL> flow-whitelist alias <GTP whitelist file alias> imsi <IMSI number> gsapp-resource <alias <alias> | all> gtp-persistence <alias <alias> | all> sip-whitelist <alias <alias> caller-id <caller ID>> stats [alias <alias> | all]**	For usage examples, refer to gsgroup. Refer to *"Viewing GigaSMART Statistics"* section in the *GigaVUE Fabric Management Guide* for more information.
show gsgroup flow-ops-report alias <alias> type flow-filtering any statistics	This command provides the information about number of active sessions and number of maximum sessions available. **Session Capacity** **Gen 2 Cards** The session capacity is

Show Command	Description
	determined from the command: **gsparam 3gpp-node-role-user 5000 stand-alone** as follows: **session limit * 1000** **Gen 3 Cards** The Gen 3 cards are completely licensed and hence the session capacity will have a constant value of **40 million** by default. **NOTE:** The maximum sessions available (session capacity) calculation is only applicable when <3gpp-node-role user> is enabled.
show gsop **alias <alias>** **all** **by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt> stats [alias <alias> [ip-frag] | all [detail] | by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt] by-gsgroup <GS group alias>>**	For usage examples, refer to gsop. Refer to the *"Viewing GigaSMART Statistics"* section in the *GigaVUE Fabric Management Guide* for more information.
show gsparams [alias <alias> | all]	For usage examples, refer to gsparams. Refer to the *"Viewing GigaSMART Statistics"* section in the *GigaVUE Fabric Management Guide* for more information.

Show Command	Description
show gmon speed-port-group	Displays all speed port groups of GigaVUE-TA25E.
show hb-profile [alias <alias> | default] [all]	For usage examples, refer to hb-profile.
show hosts	For usage examples, refer to hostname.
show ib-pathway traffic-rate {all | alias <ib_name>}	Displays the Rx and Tx statistics for either all the inter-broker pathway or a specified inter-broker pathway alias that are part of the inline flow deployment. Refer to ib-pathway.
show images	For usage examples, refer to image.
show ingress-vlan-tag	For usage examples, refer to port.
show inline-netlag traffic-rate [all | alias <alias_ name>]	Displays the Rx and Tx statistics for either all the inline netlags or a specified inline netlag alias that are part of the inline flow deployment.
show inline-network [alias <alias> | all | brief]	For usage examples, refer to inline-network.
show inline-network-group [alias <alias> | all]	For usage examples, refer to inline-network-group.
show inline-serial [alias <alias> | all]	For usage examples, refer to inline-serial.
show inline-tool [alias <alias> | all] [brief] [vlan-mapping]	For usage examples, refer to inline-tool.
show inline-tool-group [alias <alias> | all]	For usage examples, refer to inline-tool-group.
show interfaces <interface name> [brief | configured]	For usage examples, refer to interface.
show ip default-gateway [static]	For usage examples, refer to ip.

Show Command	Description
dhcp **filter [all | configured]** **route [static]**	
show ip interface [alias | all | stats]	For usage examples, refer to ip interface.
show apps icap server all | <alias>	For usage examples, refer to apps icap.
show arp	Displays all ARP information, including both static and dynamic entries.
show ipv6 neighbors	Displays all IPv6 neighbor information, which is part of the IP interface configuration. (The configuration is done as part of the IP interface in GigaVUE-OS with IPv6).
show ipv6 **default-gateway [static]** **dhcp** **filter [all | configured]** **neighbors [static]** **route [static]**	For usage examples, refer to ipv6.
show jobs [job ID]	For usage examples, refer to job.
show ldap	For usage examples, refer to ldap.
show license [box-id <box ID>]	For usage examples, refer to license.
show load-balance port-group stats <alias> <port-group name> | all>	For usage examples, refer to port-group.
show log **continuous [matching <reg exp> | not matching <reg exp>]** **files <file number> [matching <reg exp> | not matching <reg exp>]** **matching <reg exp>** **not matching <reg exp>**	For usage examples, refer to logging.

Show Command	Description
show logging	For usage examples, refer to logging.
show logging port	Displays the minimum severity of certain port log messages.
show map **access** **alias <alias>** **all** **assignment [alias <alias> | all]** **brief** **mode** **priority alias <alias>]** **stats <alias <alias> [rule <rule ID>] | all>**	For usage examples, refer to map.
alias <alias> | all]	For usage examples, refer to map.
alias <alias> | all | brief]	For usage examples, refer to map-group.
alias <alias> | all | brief]	For usage examples, refer to map-passall.
alias <alias> | all | brief]	For usage examples, refer to map-scollector.
show memory	For usage examples, refer to write.
show nhb-profile [alias <alias> | all]	For usage examples, refer to nhb-profile.
show notifications	For usage examples, refer to notifications.
show ntp [configured]	For usage examples, refer to ntp.
show pld [slot <slotID>]	For usage examples, refer to pld.
show policy [alias <alias> | detail]	For usage examples, refer to policy.
show proxy-server profile [all | alias <alias-name>]	For usage examples, refer to hb-profile.

Show Command	Description
show port access [all | box-id <box ID> [brief] | port-list <port list> [brief] | slot <slot ID> [brief]] alias assignment [all | box-id <box ID> [brief] | port-list <port list> [brief] | slot <slot ID> [brief]] comment discovery [all | box-id <box ID> [brief] | brief [port-list <port-list>] | port-list <port list> [brief] | slot <slot ID> [brief]] faceplate-number-mapping port-list <port list> filter [port-list <port list>] in-use [type <network | tool | hybrid | inline-network | inline-tool | gs>] mode [box-id <box ID> | slot <slot ID>] params all box-id <box ID> [in-use] | [brief] | [type] | [unused] brief [port-list <port list>] in-use [type <network | tool | hybrid | inline-network | inline-tool | gs>] port-list <port list> [brief] slot <slot ID> [brief] | [in-use] | [type] | [unused] type <network | stack | tool | hybrid | inline-network | inline-tool | gs> unused [type <network | stack | tool | hybrid | inline-network | inline-tool | gs>] port-number-mapping port-list <port list> stats [all | box-id <box ID> | port-list <port list> | slot <slot ID>] threshold [drop | error] [all | box-id <box ID> | port-list <port list>] tool-share [all | box-id <box ID> | port-list <port list> | slot <slot ID>] type <network | stack | tool | hybrid | inline-network | inline-tool | gs> unused [type <network | stack | tool | hybrid | inline-network | inline-tool | gs>] utilization [all | box-id <box ID> | port-list <port list> | slot <slot ID>]	For usage examples, refer to port.

Show Command	Description
show port-group [alias <alias> | all | brief]	For usage examples, refer to port-group.
show port-pair [alias <alias> | all | brief]	For usage examples, refer to port-pair.
show profile <current | history> buffer [port-list <min | hour | day | week | cur>] | <all> port [port-list <min | hour | day | week | cur>] | <all> utilization [port-list <min | hour | day | week | cur>] | <all>	For usage examples, refer to card (GigaVUE-OS® HC Series) and port.
show pseudo-slot portmap [box-id <box ID>]	For usage examples, refer to filter-template.
show ptp [box-id <box ID> | configured]	For usage examples, refer to ptp.
show radius	For usage examples, refer to radius-server.
[alias <alias> | all]	For usage examples, refer to redundancy-profile.
show roles [assignment] [alias <alias> | all]	Displays the currently configured roles.
show running-config [full]	Displays commands to recreate current running configuration.
show serial	For usage examples, refer to serial.
show snmp [engineID | events | host]	For usage examples, refer to snmp-server.
[alias <alias> | all | brief]	For usage examples, refer to spine-link.
show ssh <client | server [host-keys]>	For usage examples, refer to ssh.
show stack-link [alias <alias> | all | brief]	For usage examples, refer to stack-link.
show system stacking-mode	For usage examples, refer to system.
show sync	For usage examples, refer to sync.

Show Command	Description
show system	For usage examples, refer to system.
show system-health [box-id <box ID> | config [box-id <box ID>] | status [box-id <box ID>] | disk activity]	For usage examples, refer to system-health.
show tacacs	For usage examples, refer to tacacs-server.
show terminal	For usage examples, refer to terminal.
show tool-mirror [alias <alias> | all | brief]	For usage examples, refer to tool-mirror.
show traffic [all | brief | port <port-list> | slot <slot ID> | vport alias <alias> | all]	Displays traffic forwarding schemes.
show sffp-profile [alias <alias> | all]	For usage examples, refer to sffp profile.
show tunnel	Reserved for future use.
show tunnel-endpoint [alias | state <port-list <GS port ID | GS group alias>> | stats <port-list <GS port ID | GS group alias>> | te-id <te ID>]	For usage examples, refer to tunnel-endpoint.
show usernames [assignment] [alias <alias> | all]	For usage examples, refer to username.
show users [history [username <username>] | roles]	For usage examples, refer to username.
show version [all | box-id <box ID> | chassis-serial <chassis serial number> | concise]	Displays version information for current system image.
show vlan-resource	Displays the VLAN resources that are used, and available. It also displays feature-wise VLAN usage.

Show Command	Description
show vport [alias <alias> | all | mmon-report <alias>| sensor-diag <alias> | stats [alias <alias> | all]]	For usage examples, refer to vport. show vport mmon-report is supported only on Generation 2 GigaSMART Module.
show web	For usage examples, refer to web.
show whoami	Displays the identity and roles of the current user.

GigaVUE V Series show

Required Command-Line Mode = Standard

Use the **show** command to view configuration and status information for GigaVUE V Series Nodes Configuration commands in the CLI .

The following table describes the **show diag** commands:

Show Command	Description
show diag	Displays diagnostic information such as pstree, lspci, lscpu, ifconfig -a, arp, ip -4 neigh show, ip -6 neigh show, apiv node, apiv appenvs, apiv arp, apiv ndv, apiv reps, apiv teps, apiv links, apiv inventory/chasis, apiv stats, route, apiv maps, apiv aeps, It also runs and displays the output for the following scripts: • apiv stats/metadata/exporters apiv stats/metadata/userDefinedApp .

sleep

Required Command-Line Mode = Enable

Required Command-Line Mode = Configure

Use the **sleep** command to pause the CLI for a specified number of seconds in order to add a delay.

For example, when you enter several configuration commands in a row, such as by copying commands from a file, you can use the **sleep** command to introduce a pause between commands. In a cluster environment, this provides time for the standby units to synchronize with the leader and thus avoid any leader/standby synchronization failures.

The **sleep** command has the following syntax:

```
sleep <number of seconds>
```

For example:

```
(config) # sleep 2
```

snmp-server

Required Command-Line Mode = Configure

Use the **snmp-server** command to configure all SNMP-related functionality on the GigaVUE-OS node, including enabling SNMP generally, adding notification destinations, specifying notification events, adding standard MIB-II contact/location info, and enabling the system's SNMP server so that management stations can poll the GigaVUE-OS node remotely using standard SNMP commands (**Get**, **GetNext**, **Walk**, and so on).

Refer to the *"Use SNMP"* chapter in the *GigaVUE Administration Guide* for details on configuring SNMP.

The **snmp-server** command has the following syntax:

```
snmp-server
    community <community string>
  contact <string>
  enable [communities] [mult-communities] [notify]
  host <IPv4 / IPv6 address or Hostname>
    disable
    informs [community] [port <port number>] [version <2c | 3>]
      <engineID <engine ID> <user <username>> <auth | encrypted auth | prompt auth>
        <md5 <password> | sha <password> <priv <des <password> | aes-128 <password>> |
sha256 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> |
sha384 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> |
sha512 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>>
    traps [community] [port <port number>] [version <1 | 2c | 3>]
      <user <username>> <auth | encrypted auth | prompt auth>
        <md5 <password> | sha <password> <priv <des <password> | aes-128 <password>> |
sha256 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> |
sha384 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> |
sha512 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>>
  location <string>
  notify
    community <string>
    event [systemreset] [configsave] [modulechange] [linkspeedstatuschange]
[unexpectedshutdown] [userauthfail] [firmwarechange] [packetdrop] [gspacketdrop]
```

[tunnelstatus] [tunneldeststatus] [bufferoverusage] [rxtxerror] [powerchange] [fanchange] [portutilization] [lowportutilization] [ibstatechange] [gscpuutilization] [gsportutilz] [gsportlowutilz] [evallicensereminder] [watchdogreset] [inlinetoolrecovery] [gdpupdate] [opticstemp] [exhausttemp] [switchcputemp] [cputemp] [2ndflashboot] [operationmode] [gigasmartcputemp] [eporttemp] [policytrigger] [process-cpu-threshold] [process-mem-threshold] [system-cpu-threshold] [system-mem-threshold] [ipgatewaystatus] [gsproxyserverconnectionstatus] [gsmobtunnelutilization] [gsmobsessionutilization] [gsmobpersistencefailure] [gsmobpacketbufferutilization] [gsmobcpuutilization] [all]

port <port number>

port <port number>

user <username | admin> v3 <auth | encrypted auth | prompt auth> <md5 <password> | sha <password>

<priv <des <password> | aes-128 <password>> | sha256 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> | sha384 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> | sha512 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>>

<enable>

throttle

event [gsproxyserverconnectionstatus interval <interval in seconds>] [gsmobpacketbufferutilization interval <interval in seconds>] [gsmobcpuutilization interval <interval in seconds>]

Notes

Refer to the following notes about SNMPv3 hosts and SNMPv3 users:

- **On upgrading GigaVUE-OS devices from software version 6.3.00 to 6.4.00**: SNMPv3 host requires SNMPv3 users to be created in the system. If SNMPv3 users are not already created in software version 6.300 or lower, then on upgrading to software version 6.4.00, the users will be created by the system based on SNMPv3 host configuration.
- **From software version 6.4 and higher (fresh installations)**: Before configuring SNMPv3 host, you must create SNMPv3 users. Otherwise, traps will not be sent. It is recommended to configure the same authentication parameters for users as well as the SNMP hosts to avoid using wrong credentials for encryption.

The following table describes the arguments for the **snmp-server** command.

Argument	Description
community <community string>	Specifies the read-only community string used to connect to this node using SNMP. The default value is public. If you enable the mult-communities option, you can specify multiple community strings for the node. Refer to the *"Recommendations for Vulnerabilities"* section in the

Argument		Description
		GigaVUE Fabric Management Guide for more information. Examples: **(config) # snmp-server community secret**
contact <string>		Specifies the MIB-II contact information for this device (syscontact). For example: **(config) # snmp-server contact "John Smith"**
enable **[communities]** **[mult-communities]** **[notify]**		Specifies different aspects of the system's SNMP functionality as follows: • **enable**—Enables the SNMP server in general. • **enable communities**—Enables community-based authentication for the system. • **enable mult-communities**—Allows the configuration of multiple communities with the **snmp-server community <community string> command** for authentication. • **enable notify**—Enables notifications, allowing SNMP informs and traps to be sent from this node/cluster. Once notifications are enabled, you can specify the events that will generate traps using the **snmp-server notify** command. Refer to the *"Recommendations for Vulnerabilities"* section in the *GigaVUE Administration Guide* for details.
host		Adds a destination for SNMP notifications.
	<IPv4 / IPv6 address or Hostname>	Specifies the IPv4 address, IPv6 address, or the hostname of this destination for SNMP traps. You can specify multiple destinations, each with its own trap version and community string.

Argument		Description
	disable	Temporarily disables the sending of traps to the specified destination. For example, the following command disables traps sent to 192.168.1.25: **(config) # snmp-server host 192.168.1.25 disable** To re-enable traps for the specified destination: **(config) # no snmp-server host 192.168.1.25 disable**
	informs [community] [port <port number>] [version <2c | 3>] **for version 3 only:** **<engineID <engine ID> <user <username>> <auth | encrypted auth | prompt auth><md5 <password> | sha <password> <priv <des <password> | aes-128 <password>> | sha256 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> | sha384 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> | sha512 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>>**	Specifies whether or not this destination should be used as a destination for SNMP inform events. You can override the default port and community string configured with the **snmp-server notify** commands using custom strings for this destination. Strings must be supplied in quotation marks. If you supply an empty community string (**“”**), the global community string specified with the **snmp-server notify community <community string>** command is used instead. Also specifies the version of SNMP to use. You can specify either version 2c or 3. The default is version 2c. For example: **(config) # snmp-server host 1.1.1.1 informs** If you specify version 3, you also supply an engine ID, user name, and other settings to be sent with the notification.
	traps [community] [port <port number>] [version <1 | 2c | 3>] **for version 3 only:** **<user <username>> <auth | encrypted auth | prompt auth> <md5 <password> | sha <password> <priv <des <password> | aes-128 <password>> | sha256 <password> <priv <des <password> | aes-128 <password> |**	Specifies whether this destination should be used as a destination for SNMP trap events. You can override the default port and community string configured with the **snmp-server notify** commands using custom strings for this destination. Strings must be supplied in

Argument		Description
	aes-256 <password>> | sha384 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>> | sha512 <password> <priv <des <password> | aes-128 <password> | aes-256 <password>>	quotation marks. If you supply an empty community string (**“”**), the global community string specified with the **snmp-server notify community <community string>** command is used instead. Also specifies the version of SNMP to use. You can specify either version 2c or 3. The default is version 2c. For example: **(config) # snmp-server host 1.1.1.1 traps** If you specify 3, you must specify user name and other settings to be sent with the notification. Refer to the Notes section for SNMPv3 hosts and SNMPv3 user configuration details during upgrade from 6.3.00 to higher versions and for fresh installation of GigaVUE-OS nodes.
location <string>		Specifies the location of the system (syslocation). For example: **(config) # snmp-server location “2nd Floor”**
notify community <community string>		Specifies the default community string to be sent with SNMP notifications. If a destination has its own community string configured with **snmp-server host**, that string takes precedence over this one. However, if a destination does not have a string configured, this value is used. For example: **(config) # snmp-server notify community public**
notify event		Enables each of the events available for SNMP notifications, as follows. For example: **(config) # snmp-server notify event all**

Argument		Description
	systemreset	Sends an SNMP notification to all configured destinations each time the system starts up, either as a result of cycling the power or a soft reset initiated by the **reload** command.
	configsave	Sends an SNMP notification to all configured destinations each time the configuration is saved to local storage (for example, by using the **write memory** command).
	modulechange	Sends an SNMP notification to all configured destinations when it detects a change in line card type from the last polling interval. This typically happens when a line card is removed from a slot or inserted in an empty slot.
	linkspeedstatuschange	Sends an SNMP notification to all configured destinations in the following situations: • Each time a port's link status changes from up to down or vice-versa. • Each time a port's speed changes. **NOTE:** The **portlinkchange** trap is **not** sent when the Management port's link status changes. **NOTE:** The link state polling interval is 1 second. If a link state change is detected during the poll, an SNMP notification is generated.
	unexpectedshutdown	Sends an SNMP notification to all configured destinations when the system is shut down unexpectedly (for example, because power was lost and subsequently restored).
	userauthfail	Sends an SNMP notification to all configured destinations each time

Argument		Description
		a user login fails.
	firmwarechange	Sends an SNMP notification to all configured destinations when there is a firmware change.
	packetdrop	Sends an SNMP notification to all configured destinations when it detects packets being dropped on a port for 30 consecutive seconds.
	acmeReqStatus	Sends an SNMP notification to all configured destinations for the status of ACME request, the valid ACME requests are acme issue, acme renew and acme revoke.
	gsappcrashnotification	Sends an SNMP notification to all configured destinations when an GIgaSMART application crashes.
	gsdumpstatus	Sends an SNMP notification to all configured destinations requesting for the GigaSMART dump status.
	gspacketdrop	Sends an SNMP notification to all configured destinations when there are packet drops on a port in the GigaSMART card for 30 consecutive seconds. The notification will include the following information: • **Hardware Name**—specifies the name of the GigaSMART card in string format • **Level**—specifies the severity level of the trap • **Description**—provides the description of the trap • **Port Name**—specifies the port name of the interface where the packet was dropped • **Counter**—provides the number of packets dropped during the 30 seconds interval
	gsproxyserverconnectionstatus	Sends a SNMP notification to all configured destinations requesting for the proxy-server connection

Argument		Description
		status.
	gsmobtunnelutilization	Sends a SNMP notification to all configured destinations when the mobility tunnel usage has exceeded the configured threshold.
	gsmobsessionutilization	Sends a SNMP notification to all configured destinations when the mobility session has crossed the configured threshold.
	gsmobpersistencefailure	Sends a SNMP notification to all configured destinations when the persistence backup file is stale or if the restore file is corrupted.
	gsmobpacketbufferutilization	Sends a SNMP notification to all configured destinations when the packet-buffer reaches the configured threshold value.
	gsmobcpuutilization	Sends a SNMP notification to all configured destinations when the CPU utilization exceeds the configured threshold value.
	gsisslresourceutilization	Sends a SNMP notification to all configured destinations when there is an Inline SSL resource utilization overload in the GigaSMART.
	tunnelstatus	Sends an SNMP notification to all configured destinations each time the tunnel gateway status changes, either from Resolved to Not Resolved or from Not Resolved to Resolved. The status might change for example, if there is an ARP failure or if a destination is not reachable.
	tunneldeststatus	Sends an SNMP notification to all configured destinations each time the tunnel destination or tunnel endpoint status changes, either from Up to Down or from Down to Up.

Argument		Description
	bufferoverusage	Sends an SNMP notification to all configured destinations each time buffer usage has exceeded its configured threshold. Refer to the *"Configure Alarm Buffer Thresholds"* section in the *GigaVUE Fabric Management Guide*.
	rxtxerror	Sends an SNMP notification to all configured destinations each time there is a packet receive (RX) or transmit (TX) error.
	powerchange	Sends an SNMP notification to all configured destinations each time the power supply status changes. **NOTE:** On the GigaVUE-HB1, this trap is not supported. When power is lost on all power module(s), the GigaVUE-HB1 is unpowered.
	fanchange	Sends an SNMP notification to all configured destinations each time the fan status changes.
	portutilization	Sends an SNMP notification to all configured trap destinations when the percentage utilization on a port rises above the high threshold configured with the **port <port list> alarm high-utilization-threshold <percentage>** command. Utilization alarms are written to syslog and forwarded to all SNMP management stations configured as trap destinations. **NOTE:** Network ports always use an Rx threshold; tool ports always use Tx. Refer to the *"Working with Port Utilization Measurements"* section in the *GigaVUE Fabric Management Guide* for more

Argument		Description
		information.
	lowportutilization	Sends an SNMP notification to all configured trap destinations when the percentage utilization on a port falls below the low threshold configured with the **port <port list> alarm low-utilization-threshold <percentage>** command. Utilization alarms are written to syslog and forwarded to all SNMP management stations configured as trap destinations. Refer to the *"Working with Port Utilization Measurements"* section in the *GigaVUE Fabric Management Guide* for more information.
	ibstatechange	Sends an SNMP notification to all configured destinations each time there is an inline bypass forwarding state change.
	gscpuutilization	Sends an SNMP notification to all configured destinations when CPU utilization on the GigaSMART engine exceeds the configured upper (rising) threshold. Refer to the "*GigaSMART CPU Utilization Statistics*" section in the *GigaVUE Fabric Management Guide* for details.
	gsportutilz	Sends an SNMP notification to all configured destinations when the GigaSMART engine port utilization reaches the high threshold value configured for the engine port.
	gsportlowutilz	Sends an SNMP notification to all configured destinations when the GigaSMART engine port utilization reaches the low threshold value configured for the engine port.
	evallicensereminder	Sends an SNMP notification to all configured destinations when an

Argument		Description
		evaluation license is about to expire. This trap is sent when there are 30, 15, 10, 5, 4, 3, 2, and 1 days remaining on the 45-day license period. Refer to the *"GigaSMART Evaluation Licenses"* section in the *GigaVUE Fabric Management Guide* for details.
	watchdogreset	Sends an SNMP notification to all configured destinations each time the watchdog monitor has to reset a failed process on the system.
	inlinenetforwardingstatechange	Sends an SNMP notification to all configured destinations each time there is an inline forwarding status change.
	inlinenetlagforwardingstatechange	Sends an SNMP notification to all configured destinations each time there is a change in the inline network lag forwarding status.
	inlinetoolgroupoperstatechange	Sends an SNMP notification to all configured destinations each time there is a change in the Inline group tool operational status.
	inlinetoolhbchange	Sends an SNMP notification to all configured destinations each time an inline tool heartbeat status changes across INIT (initializing), NORMAL, LOST, RECOVERING states.
	inlinetooloperstatechange	Sends an SNMP notification to all configured destinations each time there is a change in the Inline tool operational status.
	inlinetoolrecovery	Sends an SNMP notification to all configured destinations each time an inline tool has recovered. The user can then manually put the inline tool back into service.
	gdpupdate	Sends an SNMP notification to all configured destinations each time

Argument		Description
		a new Gigamon discovery neighbor is discovered or Gigamon discovery information for an existing neighbor is changed or expired.
	opticstemp *	Sends an SNMP notification to all configured destinations each time the temperature of the GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E optics (transceivers) reach warning, alert, and critical thresholds. Refer to the device Hardware Installation Guide for details.
	exhausttemp *	Sends an SNMP notification to all configured destinations each time theGigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E ambient temperature reaches warning, alert, and critical thresholds. Refer to the device Hardware Installation Guide for details.
	netflowoutofresource	Sends an SNMP notification to all the configured destinations each time when the resource allocation fails in the NetFlow application.
	switchcputemp *	Sends an SNMP notification to all configured destinations each time the temperature of the GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E switch CPU reaches warning, alert, and critical thresholds.. Refer to the device Hardware Installation Guide for details.

Argument		Description
	cputemp *	Sends an SNMP notification to all configured destinations each time the temperature of the GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E system boots from the second flash. Refer to the device Hardware Installation Guide for details.
	2ndflashboot *	Sends an SNMP notification to all configured destinations each time theGigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E system boots from the second flash. Refer to the device Hardware Installation Guide for details.
	operationmode	Sends an SNMP notification to all configured destinations each time a node in a cluster changes from operational mode to safe mode or from operational mode to limited mode. Refer to the *"Cluster Safe and Limited Modes"* section in the *GigaVUE Fabric Management Guide* for details.
	gigasmartcputemp	Sends an SNMP notification to all configured destinations each time the temperature of the GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3, GigaSMART engine reaches warning, alert, and critical thresholds. Refer to the device Hardware Installation Guide for details.
	eporttemp *	Sends an SNMP notification to all configured destinations each time the temperature of the GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3 GigaSMART engine

Argument		Description
		ports (e1 and e2) reach warning, alert, and critical thresholds. Refer to the device Hardware Installation Guide for details.
	vportstatechange	Sends an SNMP notification to all configured destinations each time a vport State changes.
	topswitchtemp	Sends an SNMP notification to all configured destinations each time the temperature sensor near the top of the DELL-S4112F-ON's switch board reaches or exceeds the threshold temperature.
	throttle-report	Sends an SNMP notification to all configured destinations each time a throttle reporting is initiated. A throttle report is generated only when the number of traps exceeds or matches the configured threshold count.
	System-fips-test	Sends an SNMP notification to all configured destinations indicateing status of FIPS test executed.
	sfpcagetemp	Sends an SNMP notification to all configured destinations each time the sfp cage board sensor's temperature reaches critical thresholds.
	qsfpcagetemp	Sends an SNMP notification to all configured destinations each time the qsfp cage board sensor's temperature reaches critical thresholds.
	rearpaneltemp	Sends an SNMP notification to all configured destinations each time the rear panel board sensor's temperature reaches critical thresholds.
	nearcputemp	Sends an SNMP notification to all configured destinations each time the cpu board sensor's

Argument		Description
		temperature reaches critical thresholds.
	cpusoctemp	Sends an SNMP notification to all configured destinations each time the temperature of the cpu core that is read from the system files reaches or exceeds the threshold temperature.
	cluster-role-change	Sends an SNMP notification to all configured destinations each time there is a cluster role change event.
	cluster-node-leave	Sends an SNMP notification to all configured destinations each time a node leaves a cluster.
	cluster-node-join	Sends an SNMP notification to all configured destinations each time a node joins a cluster.
	bottomswitchtemp	Sends an SNMP notification to all configured destinations each time the temperature sensor near the bottom of the DELL-S4112F-ON's switch board reaches or exceeds the threshold temperature.
	bcmsoctemp	Sends an SNMP notification to all configured destinations each time the temperature sensor of the hardware reaches or exceeds the threshold temperature.
	Sublicensereminder	Sends an SNMP notification to all configured destinations when a sublicense is about to expire. The notification contains all the details like box, slot, features, and days for the license to expire.
	licensereminder	Sends an SNMP notification to all configured destinations when a license is about to expire or already has expired. The notification contains all the details like box, slot, features, and days for the license to expire.

Argument		Description
	lpgwstatusupdate	Sends an SNMP notification to all configured destinations each time IP Gateway status is updated.
	cluster-join-timeout	Sends an SNMP notification to all configured destinations each time a node tries to join a cluster and fails.
	policytrigger	Sends an SNMP notification to all configured destinations each time a policy is triggered. Refer to the *"Configuring Active Visibility"* section in the *GigaVUE Fabric Management Guide* for details.
	process-cpu-threshold *	Sends an SNMP notification to all configured destinations each time the control card memory utilization of GigaVUE-HC1, GigaVUE-HC1-Plus, GigaVUE-HC3,GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-OS-TA100, , GigaVUE-TA200, or GigaVUE-TA200E exceed the pre-configured process threshold values. Refer to the device Hardware Installation Guide for details.
	process-mem-threshold	Sends an SNMP notification to all configured destinations each time the control card memory utilization exceeds the pre-configured process threshold values.
	system-cpu-threshold	Sends an SNMP notification to all configured destinations each time the control card CPU utilization exceeds the pre-configured system threshold values.
	system-mem-threshold	Sends an SNMP notification to all configured destinations each time the control card memory utilization exceeds the pre-configured system threshold values.
	ipgatewaystatus	Sends an SNMP notification to all configured destinations each time

Argument		Description
		the ARP/NDP status change.
	all	Specifies that all SNMP notification events be sent.
	psumismatch	Sends an SNMP notification to all configured destinations each time when the different code PSUs inserts in the same chassis.
notify port <port number>		Specifies the default port to which SNMP notification events should be sent. The GigaVUE HC Series node sends SNMP notification events to this port on all configured destinations without an explicit port override configured with the **snmp host** arguments. For example: **(config) # snmp-server notify port 123**
port <port number>		Specifies the UDP port to be used for the system's SNMP server. The default is 161. For example: **(config) # snmp-server port 123** Refer to the *"Recommendations for Vulnerabilities"* section in the *GigaVUE Administration Guide*.
event <trap> interval <1 and 86400 > [report-threshold <0 and 2147483647>]		Configures SNMP trap throttling and specifies the trap that has to be throttled as follows: • **event**—Specifies the events that has to be throttled. Type **throttle event** ? to see the list of available events. • **interval**—Configures the throttling time interval. • **report-threshold**—Configures the threshold count to enable throttle reporting. Sents the report only when the number of traps exceeds or matches the configured threshold count.
user <username | admin> v3		Configures per-user security

Argument		Description
		parameters for SNMP v3 access to the system, as follows.
	enable	Enables or disables SNMP v3 access for the specified username.

Argument		Description
	<auth | encrypted auth | prompt auth>	Specifies arguments that work together to configure passwords for SNMP v3 access for the specified user, including the hash algorithm. The order of the arguments after the **v3** keyword governs how the passwords are entered, as follows: **NOTE:** The MD5 authentication protocol is not compatible with FIPS mode. Use SHA or SHA-2 authentication algorithms instead. • If **auth** is the next word after **v3**, the passwords are specified in plaintext on the command-line. • If **encrypted** is the next word after **v3**, the passwords are specified encrypted (hashed) on the command-line. • If **prompt** is the next word after **v3**, the passwords are not specified on the command-line. Instead, you are prompted for them while the command is executing. • If the optional **priv** argument is not included, only the auth password is prompted for. • If the optional **priv** argument is included, you are prompted for the privacy password. If you supply an empty string at this prompt, the same password as specified for authentication is used. Refer to the Notes section for SNMPv3 hosts and SNMPv3 user configuration details during upgrade from 6.3.00 to higher versions and for fresh installation ofGigaVUE-OS nodes.
	<md5 <password> | sha <password> | sha256 <password> | sha384 <password> | sha512 <password>>	
	<priv <des <password> | aes-128 <password> | aes-256 <password>>	
throttle	**gsproxyserverconnectionstatus interval**	Configures SNMP trap throttling time interval for the proxy server configuration status.

Argument		Description
	gsmobpacketbufferutilization interval	Configures SNMP trap throttling time interval for the packet buffer utilization.
	gsmobcpuutilization interval	Configures SNMP trap throttling time interval for the CPU utilization.

* Though this trap is available to be configured in all the platforms, it works only in the specified hardware platforms.
The traps where no hardware platform is specified work on all platforms.

NOTE: Throttle support is not available for gsmobtunnelutilization ,gsmobsessionutilization and gsmobpersistencefailure events.

Related Commands

The following table summarizes other command related to the **snmp-server** command:

Task	Command
Displays SNMP configuration information.	**# show snmp**
Displays the SNMP engine ID for the local system.	**# show snmp engineID**
Displays the events for which SNMP traps will be sent.	**# show snmp events**
Displays SNMP host settings.	**# show snmp hosts**
Displays SNMP throttle	**# show snmp throttle**
Deletes all SNMP communities and reset to the default community (public).	**(config) # no snmp-server community**
Deletes a specified SNMP community.	**(config) # no snmp-server community secret**
Deletes the contact information.	**(config) # no snmp-server contact**
Disables community-based authentication.	**(config) # no snmp-server enable communities**
Allows only a single community to be configured.	**(config) # no snmp-server enable mult-communities**
Disables the sending of SNMP notifications (traps and informs).	**(config) # no snmp-server enable notify**
Deletes this host by its IPv4 or IPv6 address.	**(config) # no snmp-server host**

Task	Command
	1.1.1.1
Re-enables the sending of all notifications to this host.	**(config) # no snmp-server host 1.1.1.1 disable**
Resets the target port for informs to the default value.	**(config) # no snmp-server host 1.1.1.1 informs port**
Resets the target port for traps to the default value.	**(config) # no snmp-server host 1.1.1.1 traps port**
Deletes the location information.	**(config) # no snmp-server location**
Resets the default notification community to the default.	**(config) # no snmp-server notify community**
Specifies that all events are not to be sent as traps.	**(config) # no snmp-server notify event all**
Disables the sending of a trap for a specified event.	**(config) # no snmp-server notify event cputemp**
Disables the throttling for a specified event	**(config) # no snmp-server throttle event**
Disables the SNMP traps notification when the GigaSMART engine port utilization reaches the high threshold value.	**(config) # no snmp-server notify event gsportutilz**
Disables the SNMP traps notification when the GigaSMART engine port utilization reaches the low threshold value.	**(config) # no snmp-server notify event gslowportutilz**
Disables throttling for all traps	**(config) # no snmp-server throttle event all**
Resets the default notifications port number to the default.	**(config) # no snmp-server notify port**
Resets the SNMP agent port to the default (161).	**(config) # no snmp-server port**
Deletes the specified SNMP v3 user.	**(config) # no snmp-server user user1 v3**
Disables all SNMP v3 access for the specified user.	**(config) # no snmp-server user user1 v3 enable**

spine-link

Required Command-Line Mode = Configure

Use the **spine-link** command to configure spine links, which are part of the configuration of the leaf and spine architecture with multiple paths for achieving high availability in a cluster environment. Refer to the *"Multi-Path Leaf and Spine"* chapter in the *GigaVUE Fabric Management Guide* for details.

The **spine-link** command has the following syntax:

```
spine-link alias <alias>
  comment <comment>
  port-list <port-list>
```

The following table describes the arguments for the **spine-link** command:

Argument	Description
alias <alias>	Specifies an alias for the spine link. For example: **(config) # spine-link alias leaf1spine**
comment <comment>	Adds a comment to a spine link. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks. For example: **(config) # spine-link alias leaf1spine comment "Leaf 1 Spine"**
port-list <port-list>	Specifies the list of stack GigaStream aliases in the spine link. Separate each alias with a comma. For example: **(config) # spine-link alias leaf1spine port-list leaf1spine1gs,leaf1spine2gs**

Related Commands

The following table summarizes other commands related to the **spine-link** command:

Task	Command
Displays all spine links.	**# show spine-link**
Displays a specific spine link.	**# show leaf1spine**
Displays all spine links.	**# show spine-link all**
Displays all spine links in table format.	**# show spine-link brief**
Deletes a specific spine link. **Note**: To delete a spine link, first delete the participating stack links.	**(config) # no spine-link alias leaf1spine**
Deletes all spine links. Type YES to confirm the deletion or type NO to cancel it.	**(config) # no spine-link allEnter 'YES' to confirm this operation:**

ssh

Required Command-Line Mode = Configure

Use the **ssh** command to enable, disable, and configure the GigaVUE-OS® HC Series node's SSH server for access to the Mgmt port.

The **ssh** command has the following syntax:

```
ssh
   client
    ciphers <aes128-cbc | aes128-ctr | aes128-gcm | aes192-ctr | aes256-cbc | aes256-ctr aes256-gcm | chacha20-poly1305@openssh.com>
      global <host-key-check <yes | no | ask> | known-host <known host entry>>
                hostkey-algo < ecdsa-sha2-nistp256 | ecdsa-sha2-nistp384 | ecdsa-sha2-nistp521 | rsa-sha2- 256 | rsa-sha2-512 >
                kex < diffie-hellman-group14-sha256 | ecdh-sha2-nistp256 | ecdh-sha2-nistp384 | ecdh-sha2-nistp521 | curve25519-sha256 | curve25519-sha256@libssh.org | diffie-hellman-group-exchange-sha256 | diffie-hellman-group16-sha512 | diffie-hellman-group18-sha512>
                macs < hmac-sha2-256 | hmac-sha2-512 | hmac-sha2-256-etm@openssh.com | hmac-sha2-512-etm@openssh.com | umac-128-etm@openssh.com>
      user <username> <authorized-key sshv2 "<public key>" | identity <rsa2 | ecdsa> <generate | private-key
       [private key] | public-key <public-key>>| known-host <known host> remove >
  server
    ciphers <aes128-cbc | aes128-ctr | aes128-gcm | aes192-ctr | aes256-cbc | aes256-ctr | aes256-gcm | chacha20-poly1305@openssh.com>
     enable
     host-key
            rsa2 <private-key [private key] | public-key <public-key>>
      ecdsa <private-key [private key] | public-key <public-key>>
       generate
                hostkey-algo < rsa-sha2-256 | rsa-sha2-512 >
                kex < diffie-hellman-group14-sha256 | ecdh-sha2-nistp256 | ecdh-sha2-nistp384 | ecdh-sha2-nistp521 | curve25519-sha256 | curve25519-sha256@libssh.org | diffie-hellman-group-exchange-sha256 | diffie-hellman-group16-sha512 | diffie-hellman-group18-sha512>
                macs < hmac-sha2-256 | hmac-sha2-512 | hmac-sha2-256-etm@openssh.com | hmac-sha2-512-etm@openssh.com | umac-128-etm@openssh.com>
     max-auth-tries
 ports <port> [port] [port] [port]..
```

The following table describes the arguments for the **ssh** command:

Argument	Description
client ciphers	Configures the ciphers to be used by the ssh client in the machine. The following ciphers are allowed in the "classic mode": • aes128-cbc * • aes128-ctr • aes128-gcm@openssh.com • aes192-ctr • aes256-cbc* • aes256-ctr • aes256-gcm@openssh.com • chacha20-poly1305@openssh.com **NOTE:** The CBC ciphers are disabled in normal mode and are available only in "secure crypto mode". You can utilize the CTR ciphers in normal mode. The following ciphers are allowed in the "secure crypto mode": • aes128-cbc • aes128-gcm@openssh.com • aes256-cbc • aes256-gcm@openssh.com The following ciphers are allowed in the "FIPS mode": • aes128-cbc • aes128-gcm@openssh.com • aes256-cbc • aes256-gcm@openssh.com
client global <host-key-check <yes | no | ask>>	Sets SSH client configuration to control how host key checking is done, as follows: • **yes**—Specifies strict host key checking, which only permits connection if a matching host key is in the known hosts file and which does not access systems without pre-configured host keys. • **ask**—Prompts the user to accept new host keys. • **no**—Specifies non-strict host key checking, which always permits connection and accepts any new or changed host keys without checking. For example: **(config) # ssh client global host-key-check yes**
client global <known-host <known host entry>>	Adds an entry to the global known-hosts configuration file.
client	Configures the hostkey algos to be used by the ssh client in the machine.

Argument	Description
hostkey-algo	The following hostkey algos are allowed in the "classic mode": • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521 • rsa-sha2-256 • rsa-sha2-512 The following hostkey algos are allowed in the "secure crypto mode": • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521 The following hostkey algos are allowed in the secure " FIPS mode": • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521
client kex	Configures the kex to be used by the ssh client in the machine. The following kex are allowed in the "classic mode": • diffie-hellman-group14-sha256 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 • curve25519-sha256 • curve25519-sha256@libssh.org • diffie-hellman-group-exchange-sha256 • diffie-hellman-group16-sha512 • diffie-hellman-group18-sha512 The following kex are allowed in the "secure crypto mode": • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 The following kex are allowed in the "FIPS mode": • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521

Argument	Description
client macs	Configures the macs to be used by the ssh client in the machine. The following macs are allowed in the "classic mode": • hmac-sha2-256 • hmac-sha2-512 • hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • umac-128-etm@openssh.com The following macs are allowed in the "secure crypto mode": • hmac-sha2-256 • hmac-sha2-512 The following macs are allowed in the "FIPS mode": • hmac-sha2-256 • hmac-sha2-512
client user <username> <authorized-key sshv2 "<public key>">	Adds the specified key to the list of authorized SSHv2 RSA or DSA public keys for this user account. An example of the command: ssh client user admin authorized-key sshv2 "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCtHVGtbdQRd6uZ/EYb896UYxBer5DweDgbmcRTdPzLb05/k5fMwPKM96PRxbWiH/YlyUb5ZZaQ6Wm1lq+fiqKjgTw8QPXDixB5JLMAl638gH5V1mnUsoZDyonknPrVa9v1YSxgAxdw3IvZMZPIRQxx5T0PC8jhRKhvguA9mNU2Su4fqCLCC/FQ9LdyRgOgXIrHB7uEsIE6wK3UIXrxf+Dgv089aMTjcC+42Sq0IT/ckflMgKAf3eL8M4Blvw7j9Dhphof6kAC9+4/zU0vy4Ks/3VbOMYU3YuG2cMFSVxyqru9YazpplnFEFhO/rfRdaGfNOhMHNdq7Y7fpOWksnkl9p1XtH8SBA09Dberwff3TwoYch0BN0hR4lLwOYNFGRO4Gedoo4BHxdtMbaoo0Ll3ScyXQjA2wlnb6pR7si3Ttac1BQRe7mf8Hqtb2/tM4GaY6HngGGgBA1PD4n8k3cw38KDzJ221lsap6OtPOPez/EfE/sw0ySal+P0AmIHIjvic= root@localhost.localdomain"
client user <username> <identity <rsa2 | ecdsa> generate>>	Generates a new identity (private and public keys) for the specified user. When the keys are generated, the private key is written to the user's .ssh directory in a file, for example, id_dsa. The rsa2 and**ecdsa**arguments specify generation of ECDSA and RSA v2 keys for SSHv2.
client user <username> <identity <rsa2 | ecdsa> <private-key [private key] | public-key <public-key>>	Specifies the public or private key (of the specified type) for the specified user. This is an alternative to generating the key. The **rsa2** and**ecdsa** arguments specify generation of RSA v2 and ECDSA keys for SSHv2. If private-key or public-key is specified without a key, the user is prompted for the key.
client user	Removes a known host from a specified user's .ssh known_hosts file.

Argument	Description
<username> <known-host <known host> remove>	
server enable	Enables the SSH server on the GigaVUE HC Series node for connections to the Mgmt port. You can also disable SSH access with the **no ssh server enable** command. For example: **(config) # ssh server enable**
server ciphers	Configures the ciphers in the ssh server running in our device. The following ciphers are allowed in the "classic mode" : • aes128-cbc • aes128-ctr • aes128-gcm@openssh.com • aes192-ctr • aes256-cbc • aes256-ctr • aes256-gcm@openssh.com • chacha20-poly1305@openssh.com The following ciphers are allowed in the "secure crypto mode": • aes128-cbc • aes128-gcm@openssh.com • aes256-cbc • aes256-gcm@openssh.com The following ciphers are allowed in the "FIPS mode": • aes128-cbc • aes128-gcm@openssh.com • aes256-cbc • aes256-gcm@openssh.com
server host-key rsa2 <private-key [private key] | public-key	Changes the SSH server host keys provided with the GigaVUE HC Series node, as follows: • **generate**—Generates new RSA and DSA host keys. • **rsa2**, or **ecdsa2**—Supplies a specific value for a public or private key of the specified type. • **private-key** or **public-key>**—Specifies whether you are generating a private key or a public key. For example, to generate new RSA and DSA host keys for SSH: **(config) # ssh server host-key generate** For example, to set a new private-key for host keys of type rsa2: **(config) # ssh server host-key rsa2 private-key**

Argument	Description
<public-key>> ecdsa2 <private-key **[private key] |** **public-key <public-key>> generate**	You will be prompted to enter the key.
server hostkey-algo	Configures the hostkey algos to be used by the ssh server in the machine. The following hostkey algos are allowed in the "classic mode": • rsa-sha2-256 • rsa-sha2-512 The following hostkey algo is allowed in the "secure crypto mode": • ecdsa-sha2-nistp384 The following hostkey algo is allowed in the "FIPS mode": • ecdsa-sha2-nistp384
server kex	Configures the kex to be used by the ssh server in the machine. The following kex are allowed in the "classic mode": • diffie-hellman-group14-sha256 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 • curve25519-sha256 • curve25519-sha256@libssh.org • diffie-hellman-group-exchange-sha256 • diffie-hellman-group16-sha512 • diffie-hellman-group18-sha512 The following kex are allowed in the "secure crypto mode": • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 The following kex are allowed in the "FIPS mode": • ecdh-sha2-nistp256 • ecdh-sha2-nistp384

Argument	Description
	• ecdh-sha2-nistp521
server macs	Configures the macs to be used by the ssh server in the machine. The following macs are allowed in the "classic mode": • hmac-sha2-256 • hmac-sha2-512 • hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • umac-128-etm@openssh.com The following macs are allowed in the "secure crypto mode": • hmac-sha2-256 • hmac-sha2-512 The following macs are allowed in the "FIPS mode": • hmac-sha2-256 • hmac-sha2-512
server max-auth-tries	Specify the maximum number of authentication attempts permitted per SSH connection. You can configure up to a maximum of 3 attempts. The default is 3. For example: **(config) # ssh server max-auth-tries 2**
server ports <port> [port] [port] [port]..	Specifies the TCP port(s) on which the SSH server listens. Multiple ports can be specified. The default is 22. For example: **(config) # ssh server ports 23**

Related Commands

The following table summarizes other commands related to the **ssh** command:

Task	Command
Displays SSH client settings.	**# show ssh client**
Displays SSH server settings.	**# show ssh server**
Displays SSH server settings with full host keys.	**# show ssh server host-keys**
Resets global SSH client host key check settings.	**(config) # no ssh client global host-key-check**
Deletes the client SSH configurations and reset to	**(config) # no ssh client**

Task	Command
the default values.	**<ciphers/kex/macs/hostkey-algo>**
Deletes the server SSH configurations and reset to the default values.	**(config) # no ssh server <ciphers/kex/macs/hostkey-algo>**
Deletes a global SSH client known host entry by host.	**(config) # no ssh client global known-host <known-host-entry>**
Deletes a public key from an authorized key list for a specified user.	**(config) # no ssh client user monitor authorized-key sshv2 <public key ID>**
Deletes all SSH client identity keys for a specified user.	**(config) # no ssh client user monitor identity**
Deletes SSH client identity keys for a specified user and for a specified type of identity.	**(config) # no ssh client user monitor identity rsa2**
Disables the SSH server.	**(config) # no ssh server enable**

stack-link

Required Command-Line Mode = Configure

Use the **stack-link** command to create a stacking connection between two GigaVUE-OS nodes in a cluster. Stack-links carry traffic from network ports on one node to tool ports on a destination node.

With out-of-band clustering, cluster control traffic is carried out-of-band on its own network over the Mgmt port (eth0). With inband clustering, cluster control traffic is carried inband through the stack-link.

NOTE: Refer to the *"Creating and Managing Clusters"* section in the *GigaVUE Fabric Management Guide* for details on configuring a cluster.

Stack-links can be constructed out of individual stack ports (for example, a 40Gb port on a PRT-H00-Q02X32 line card), or, more commonly, stack GigaStream. You decide which to use with the **gigastream** and **ports** arguments in the **stack-link** command. For example, the following command creates a stack-link between the q1 40Gb port on box 1/slot 1 and the q2 port on box 2/slot 7:

(config) # stack-link alias biglink between ports 1/1/q1 and 2/7/q2 comment "40Gb Stack"

Stack links are supported at speeds of 10Gb, 40Gb, and 100Gb. Refer to the *Hardware Installation Guide* for each GigaVUE-OS node for information on stack link support.

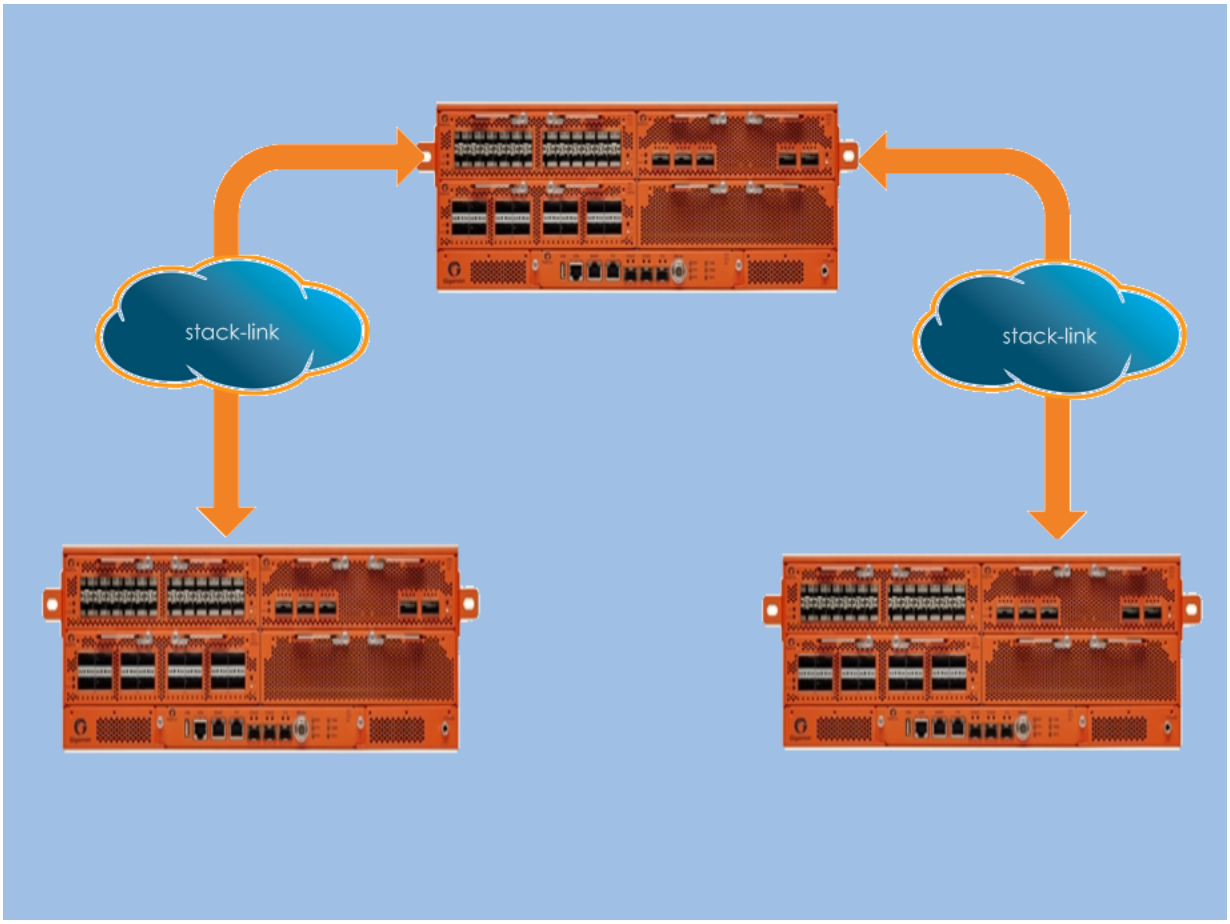
Keep in mind that because of the10Gb port density offered by the GigaVUE-OS nodes, using only one 10Gb port for a stack-link could cause a serious bottleneck. A stack GigaStream dramatically increases the bandwidth available for stack-links, letting you connect

GigaVUE-OS nodes in a cluster and still take advantage of the 10Gb port density. Alternatively, nodes with 40Gb or 100Gb ports can take advantage of their high bandwidth for stack-links.

When using stack GigaStream for stack-links, you must create a stack GigaStream on each side of the stack-link and each side must consist of the same number of ports running at the same speed.

The **stack-link** command is also used as part of the configuration of the leaf and spine architecture with multiple paths for achieving high availability in a cluster environment. Refer to the *"Multi-Path Leaf and Spine"* chapter in the *GigaVUE Fabric Management Guide* for details.

The following figure shows a simple cluster with stack-links between the nodes.



The **stack-link** command has the following syntax:

```
stack-link alias <stack alias>
   between <gigastreams <stack-link gigastream> and <stack-link gigastream>> |
      <<ports <stack-link port> and <peer stack-link port>>
   comment <comment>
```

The following table describes the arguments for the **stack-link** command:

Argument	Description
stack-link alias <stack alias>	Specifies an alias for the stack-link.
between <gigastreams <stack-link gigastream> and <stack-link gigastream>> | <<ports <stack-link port> and <peer stack-link port>>	Specifies the two sides of the stack-link, either between GigaStream or between ports. For example: **(config) # stack-link alias biglink between ports 1/1/q1 and 2/7/q2** You can also edit regular stack GigaStreams that are configured on either sides of a stack link. When a stack GigaStream is attached to a map, you can add or delete stack ports from the stack GigaStream. For example: **(config) # stack-link alias stkN2toN4 between gigastreams N2toN4 port-list 2/1/c2x1..c2x3 and N4toN2 port-list 4/1/x1..x3**
comment <comment>	Adds a comment to a stack-link. Comments can be up to 128 characters. Comments longer than one word must be enclosed in double quotation marks for example: **(config) # stack-link alias biglink comment "40Gb Stack"**

Related Commands

The following table summarizes other commands related to the **stack-link** command:

Task	Command
Displays all stack-link connections.	**# show stack-link**
Displays a specified stack-link connection.	**# show stack-link alias hc3-to-ta10-4**
Displays all stack-link connections.	**# show stack-link all**
Displays all stack-link connections in a table format.	**# show stack-link brief**
Deletes a specified stack-link.	**(config) # no stack-link alias hc3-to-ta10-4**
Deletes all stack-links.	**(config) # no stack-link all**

system

Required Command-Line Mode = Enable

Use the **system** command to restart or expedite the relaunching of individual system processes, enable secure cryptography mode, secure passwords mode, or configure arp/ndp refresh interval on the GigaVUE-OS node.

The **system** command has the following syntax:

```
system

  process <process name>
    clusterd restart
    httpd restart
    ntpd restart
    restapid restart
    snmpd restart
    sshd restart
    ugwd restart
    wsmd restart
  tls min-version <tlsversion>
    cipher-suites
    groups
  security crypto enhanced
  security fips
  security legacy
  security log martian
  security passwords
    enhanced
    min-length <length in characters>
  arp refresh-interval
  ndp refresh-interval
  stacking-mode legacy
  ztp
```

The following table describes the arguments for the **system** command:

Argument	Description
<process name>	Specifies the system process name.
clusterd restart	Restarts the clustering daemon (clusterd) process or expedites the relaunching of this process. For example: **(config) # system process clusterd restart** **NOTE:** This command only applies to cluster control. It does not affect traffic distribution.
httpd restart	Restarts the HTTP server daemon (httpd) process or expedites the relaunching of this process. For example: **(config) # system process httpd restart**

Argument	Description
ntpd restart	Restarts the NTP daemon (ntpd) process or expedites the relaunching of this process. For example: **(config) # system process ntpd restart**
restapid restart	Restarts the REST API daemon (restapid) process or expedites the relaunching of this process. For example: **(config) # system process restapid restart**
snmpd restart	Restarts the SNMP agent daemon (snmpd) process or expedites the relaunching of this process. For example: **(config) # system process snmpd restart**
sshd restart	Restarts the SSH daemon (sshd) process or expedites the relaunching of this process. For example: **(config) # system process sshd restart**
ugwd restart	Restarts the Unified Gateway daemon (ugwd) process or expedites the relaunching of this process. For example: **(config) # system process ugwd restart**
wsmd restart	Restarts the Web Session Manager daemon (wsmd) process or expedites the relaunching of this process. For example: **(config) # system process wsmd restart**
<tls> min-version	Configure the TLS minimum version. **(config) # system tls min-version <tlsversion>** For example: (config) # system tls min-version tls1.2 Refer to Web for enabling TLS Cipher Suites.
cipher-suites	Configure the TLS cipher-suites. **(config) # system tls min-version tls1.2 cipher-suites** For example: (config) # system tls min-version tls1.2 cipher-suites TLS_CHACHA20_POLY1305_SHA256
groups	Configure the TLS groups. **(config) # system tls min-version tls1.2 groups** For example: (config) # system tls min-version tls1.2 group secp384r1
security crypto enhanced	Enables the secure cryptography mode, which provides enhanced security on the management interface of the GigaVUE-OS node. For the secure cryptography mode to take effect, reload the GigaVUE-OS node or cluster. For example: **(config) # system security crypto enhanced** **(config) # reload** or **(config) # system security crypto enhanced**

Argument	Description
	(config) # cluster reload Refer to the *"Configuring Secure Cryptography Mode"* section in the *GigaVUE Administration Guide* for details. **IMPORTANT**: TLS version 1.2 is required for secure cryptography mode. When enabling secure cryptography mode, TLS version 1.2 is enabled by default. If you disable secure cryptography mode and want to change the TLS version, use GigaVUE-OS CLI command: **web server ssl min-version tls<version>**.
security fips	Enables the secure FIPS mode, ensuring compliance with Federal Information Processing Standards (FIPS) 140-3. This mode enhances system security by enforcing cryptographic standards. For example: **(config) # system security fips** A system reload is required for the change to take effect. Refer to the *"FIPS Compliance in GigaVUE-OS"* section in the *GigaVUE Administration Guide* for details.
security legacy	In legacy mode, the following algorithms are enabled in addition to the algorithms in the classic mode: **KexAlgorithms ecdh-sha2-nistp256** **ecdh-sha2-nistp384** **ecdh-sha2-nistp521** **diffie-hellman-group14-sha256** **diffie-hellman-group-exchange-sha256** **diffie-hellman-group14-sha1** **MACs hmac-sha2-512** **hmac-sha2-256** **hmac-sha1** **MACs hmac-sha2-512,hmac-sha2-256,hmac-sha1** By default the device is in classic mode, the following algorithms are enabled: **KexAlgorithms ecdh-sha2-nistp256** **ecdh-sha2-nistp384** **,ecdh-sha2-nistp521** **diffie-hellman-group14-sha256** **MACs hmac-sha2-512** **hmac-sha2-256** **MACs hmac-sha2-512,hmac-sha2-256**

Argument	Description
	Note: This configuration is allowed only when system is running in the classic mode. To check the system mode use **show system security** command.
security log martian	Enables the Martian logs that appear in the kernel logs to also appear in the syslog messages in the device.
security passwords enhanced	Enables the secure passwords mode, which increases the security of passwords on the GigaVUE-OS node. By default it is disabled. When the secure password mode is enabled, the new password must differ from the old password and contain at least eight characters. For example: **(config) # system security passwords enhanced**
security passwords min-length <length in characters>	Use **min-length** to set the minimum password length, from 8 to 64 characters. The default is 8 characters. For example: **(config) # system security passwords min-length 20** Refer to Configure Secure Password for details.
stacking-mode legacy	Selects the legacy mode for stacking. For example: **(config) # system stacking-mode legacy** Selects the default stacking mode. **NOTE:** After user confirmation the system stacking-mode legacy command immediately resets the traffic configuration and initiates a cluster reload of all nodes. After the cluster is up, the configuration saved in the backup file must be applied manually to restore the traffic configuration. For example: **[cluster: leader] (config) # no system stacking-mode legacy** **! WARNING: Changing stacking mode will automatically** **! - Take backup of config in file stacking_mode_config_backup.txt** **! - Reset factory only traffic config** **! - RELOAD the cluster** **! - User must manually apply stacking_mode_config_backup.txt after bootup** **Confirm stacking mode change? [no] YES** **Configuration saved to database 'initial'** **System shutdown initiated -- logging off.** **# after the cluster is up and user is logged back in, apply the saved configuration in the backup file:** **[cluster: leader] (config) # configuration text file stacking_mode_config_backup.txt apply fail-continue** **(config) # no system stacking-mode legacy** By default the system would disable the **stacking-mode legacy** parameter.

Argument	Description
arp refresh-interval <3~30>	Specifies the Address Resolution Protocol (ARP) refresh time interval. The timer is configurable from 3 to 30 seconds. The default is 30 seconds. When an IP interface is configured, ARP requests are sent out on the IP interface associated with tool port to find the gateway MAC address, When Tunnel encapsulation GSOP Map is configured with destination tool in local network, ARP requests are sent to the IP interface to find the tool MAC address. In response, the gateway and local tool sends an ARP reply and the control card tries to match the IP interface's IP address with the IP address of the received ARP message. If a match is found, the ARP status changes to resolved (otherwise, the ARP status is not resolved). Once ARP is resolved, this tunnel ARP timer controls the interval at which an ARP request is sent to the gateway as well as to the local tool to detect if the gateway and local tool is reachable or not. For example: **(config) # system arp refresh-interval 30** Use the **show system arp** command to display the ARP refresh interval.
ndp refresh-interval <3~30>	Specifies the Neighbor Discovery Protocol (NDP) refresh time interval. The timer is configurable from 3 to 30 seconds. The default is 30 seconds. When an IP interface is configured, Neighbor Solicitation (NS) packets are sent out on the IP interface associated with tool port to find the gateway MAC address, and Neighbour Solicitation (NS) packets are sent out on the IP interface to find the local tool address. In response, the gateway sends an Neighbor Advertisement (NA) packet and the control card tries to match the IP interface's IP address with the IP address of the received NA message. If a match is found, the IPv6 neighbor status changes to resolved (otherwise, the IPv6 Neighbor status is not resolved). Once IPv6 Neighbor is resolved, this tunnel NDP timer controls the interval at which an NS packet is sent to the gateway as well as to the local tool to detect if the gateway and local tool is reachable or not. For example: **(config) # system ndp refresh-interval 30** Use the **show system ndp** command to display the NDP refresh time interval.
ztp	Use this command to enable or disable Zero Touch Provisioning (ZTP). ZTP is enabled by default when the device undergoes a factory reset or fresh installation. To enable ZTP: **(config) # system ztp enable** To disable ZTP: **(config) # no system ztp enable** Refer to the *"Zero Touch Provisioning (ZTP)"* section in the *GigaVUE Administration Guide* for details.

Related Commands

The following table summarizes other commands related to the **system** command:

Task	Command
Displays system information.	**# show system**
Displays the stacking mode information.	**# show system stacking-mode**
Disables enhanced cryptography mode. For the change in the enhanced cryptography mode to take effect, reload the GigaVUE-OS node or cluster.	**(config) # no system security crypto enhanced** **(config) # reload** **or** **(config) # no system security crypto enhanced** **(config) # cluster reload**
Disables the secure passwords mode. Also disables the minimum length for passwords.	**(config) # no system security passwords enhanced**
Disables the management port's legacy cryptography mode, and enables the new Classic Mode security.	**(config) # no system security legacy**
Disables the secure FIPS mode.	**(config) # no system security fips**
Enables the default stacking mode and disables the legacy stacking mode.	**(config) # no system stacking-mode legacy**
Disables Zero Touch Provisioning (ZTP)	**(config)# no system ztp enable**

system-health

Use the **system-health** command to enable system health threshold checks for a specified node or for each node in a cluster.

The **system-health** command has the following syntax:

```
system-health
  box-id <box ID> threshold enable
  threshold enable
```

The following table describes the arguments for the **system-health** command:

Argument	Description
box-id <box ID> threshold enable	Enables system health threshold checks on a specified node. When enabled, SNMP events are triggered when pre-defined threshold values are exceeded for system CPU and memory utilization. For example: **(config) # system-health box-id 1 threshold enable**
threshold enable	Enables system health threshold checks. When enabled, SNMP events are triggered when pre-defined threshold values are exceeded for system CPU and memory utilization. For example: **(config) # system-health threshold enable**

Related Commands

The following table summarizes other commands related to the **system-health** command:

Task	Command
Displays cluster-wide system health information.	**# show system-health**
Displays system health information for a specified box ID.	**# show system-health box-id 1**
Displays system health configuration.	**# show system-health config**
Displays system health configuration for a specified box ID.	**# show system-health config box-id 1**
Displays system health events.	**# show system-health status**
Displays system health events for a specified box ID.	**# show system-health status box-id 1**
Displays system health disk activity information.	**# show system-health disk activity**
Disables system health threshold check for a specified box ID and disables associated SNMP events.	**(config) # no system-health box-id 2 threshold enable**
Disables system health threshold check and disables associated SNMP events.	**(config) # no system-health threshold enable**

tacacs-server

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **tacacs-server** command to specify the TACACS+ servers to be used for authentication. You can specify multiple TACACS+ servers. Servers are used as fallbacks in the same order they are specified—if the first server is unreachable, the second is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

Refer to Add a TACACS+ Server for examples of adding and configuring a TACACS+ server.

The **tacacs-server** command has the following syntax:

```
tacacs-server
  extra-user-params roles enable
  host <IPv4/IPv6 address or hostname>
    [auth-port <port number>
    auth-type <ascii | pap>
    enable
    shared-secret <string> | 9 <encrypted string>
    prompt-secret
    retransmit <retries>
    timeout <seconds>]
  shared-secret <string> | 9 <encrypted string>
```

retransmit <retries>
service <gigamon | shell>
timeout <seconds>

The following table describes the arguments for the **tacacs-server** command. The **key**, **retransmit**, and **timeout** values can be specified both globally and on a per-host basis. The values specified on the host will override the global values.

Argument		Description
extra-user-params roles enable		Enables the GigaVUE HC Series node to accept user roles assigned in the TACACS+ server. Note that the role name must match a role configured on the local node/cluster. Refer to aaa for details. The default is disabled (no roles). For example: **(config) # tacacs-server extra-user-params roles enable**
host <IPv4/IPv6 address or hostname>		Specifies the IP address (IPv4 or IPv6) or the hostname of the TACACS+ server. The same IP address can be used for more than one TACACS+ server so long as they use different **auth-port** values. Examples: **(config) # tacacs-server host 192.168.0.93** **(config) # tacacs-server host 2001:db8:a0b:12f0::11 key gigamon enable** **(config) # tacacs-server host www.MyCo.com**
	auth-port <port-number>	Specifies the UDP port number on which the TACACS+ server is running. If included, the **auth-port** must be specified immediately after the host IP address. If not specified, the port is set to the default TACACS+ port number of 49. For example: **(config) # tacacs-server host 192.168.0.93 auth-port 50**
	auth-type <ascii | pap>	Specifies whether this TACACS+ server uses ASCII or PAP authentication. The default is PAP. For example: **(config) # tacacs-server host 192.168.0.93 auth-type ascii**
	enable	Administratively enables the TACACS+ server. For example: **(config) # tacacs-server host 192.168.0.93 enable**
	shared-secret <string> | 9 <encrypted string>	Specifies the shared secret text string to be used for encryption of authentication packets sent between the GigaVUE HC Series node and this specific TACACS+ server. The key specified here overrides the global value specified in the **tacacs-server shared secret** command. For example: **(config) # tacacs-server host 192.168.0.93 shared-secret mykey2** To configure an encrypted shared-secret, use 9 keyword. **(config) # tacacs-server host 1.1.1.1 shared-secret 9**

Argument		Description
		8IEiQIIyQ1JhfkfhFBgAdn4iHdfzqPlX5tMll6oE3+Y=
	prompt-secret	Requires the user to enter the shared secret text string during login. This option is mutually exclusive with the **shared-secret** option. For example: **(config) # tacacs-server host 192.168.0.93 prompt-secret**
	retransmit <retries>	Specifies the number of times the GigaVUE HC Series node will attempt to authenticate with this specific TACACS+ server. The retransmit value specified here overrides the global value specified in the **tacacs-server retransmit** command. The default is 1. The range is from 0 to 5. Use 0 to disable retransmissions. For example: **(config) # tacacs-server host 192.168.0.93 retransmit 3**
	timeout <seconds>	Specifies how long the GigaVUE HC Series node should wait for a response from this specific TACACS+ server to an authentication request before declaring a timeout failure. The timeout value specified here overrides the global value specified in the **tacacs-server timeout** command. The default is 3 seconds. The range is from 0 to 60 seconds. For example: **(config) # tacacs-server host 192.168.0.93 timeout 45**
shared-secret <string> | 9 <encrypted string>		Specifies a global shared secret text string to be used for encryption of authentication packets sent between the GigaVUE HC Series node and all TACACS+ servers. This key can be overridden with the key specified in the **tacacs-server host** command. For example: **(config) # tacacs-server shared-secret mykey** To configure an encrypted shared-secret, use 9 keyword. **(config) # tacacs-server shared-secret 9 8IEiQIIyQ1JhfkfhFBgAdn4iHdfzqPlX5tMll6oE3+Y=**
retransmit <retries>		Specifies a global value for the number of times the GigaVUE HC Series node will attempt to authenticate with a TACACS+ server. This retransmit value can be overridden with the retransmit value specified in the **tacacs-server host** command. The default is 1. The range is from 0 to 5. Zero (0) disables retransmissions. For example: **(config) # tacacs-server retransmit 5**
service <gigamon | shell>		Specifies the authorization service that will be used for TACACS. By default, this is set to **shell**, which works for Cisco ACS 3.x. You must set it to **gigamon** for successful integration with Cisco ACS 5.3 or later. The **gigamon** setting also works for ACS 3.x. This is a global command. For example:

Argument	Description
	(config) # tacacs-server service gigamon
timeout <seconds>	Specifies a global value for how long the GigaVUE HC Series node should wait for a response from the TACACS+ server to an authentication request before declaring a timeout failure. This timeout value can be overridden with the timeout value specified in the **tacacs-server host** command. The default is 3 seconds. The range is from 0 to 60 seconds. For example: **(config) # tacacs-server timeout 30**

Related Commands

The following table summarizes other commands related to the **tacacs-server** command:

Task	Command
Displays TACACS+ servers and settings.	**# show tacacs**
Disables handling of extra user parameters sent from the TACACS+ server.	**(config) # no tacacs-server extra-user-params roles enable**
Deletes a TACACS+ host with the specified IPv4 or IPv6 address, or hostname.	**(config) # no tacacs-server host 1.1.1.1** **(config) # no tacacs-server host www.MyCo.com**
Deletes a TACACS+ host on a specified port.	**(config) # no tacacs-server host 1.1.1.1 auth-port 234**
Administratively disables the TACACS+ host.	**(config) # no tacacs-server host 1.1.1.1 auth-port 234 enable**
Administratively disables the TACACS+ host on the default port.	**(config) # no tacacs-server host 1.1.1.1 enable**
Deletes the global TACACS+ server shared secret.	**(config) # no tacacs-server shared-secret**
Resets the global TACACS+ server retransmit count to the default.	**(config) # no tacacs-server retransmit**
Resets the global TACACS+ server timeout settings to the default.	**(config) # no tacacs-server timeout**

terminal

Required Command-Line Mode = Configure

Use the **terminal** command to resize the terminal output to specified dimensions.

NOTE: The functionality provided by the **terminal** command is also available with cli.

The **terminal** command has the following syntax:

```
terminal
  length <number of lines>
  resize
  type <ansi | console | dumb | linux | screen | vt52 | vt100 | vt102 | vt220 | xterm>
  width <number of characters>
```

The following table describes the arguments for the **terminal** command:

Argument	Description
length <number of lines>	Specifies an override of the auto-detected length of the terminal. Specify the **length** in number of lines. For example: **(config) # terminal length 80**
resize	Specifies a reset of the terminal dimensions to the current window. For example: **(config) # terminal resize**
type <ansi | console | dumb | linux | screen | vt52 | vt100 | vt102 | vt220 | xterm>	Sets the terminal dimensions to a specific terminal type. For example: **(config) # terminal type xterm**
width <number of characters>	Specifies an override of the auto-detected width of the terminal. Specify the **width** in number of characters.For example: **(config) # terminal width 40**

Related Commands

The following table summarizes other commands related to the **terminal** command:

Task	Command
Displays terminal parameters.	**# show terminal**
Clears the terminal type.	**(config) # no terminal type**

threshold

Required Command-Line Mode = Configure

Use the **threshold** command to configure port-packet drop settings globally for all the ports in a device or for a given port type. The **threshold** command has the following syntax:

threshold rx/tx drop/error <global | circuit | hybrid | inline-net | inline-tool | network | stack | tool | gs> count/percent value <value>

For more information, refer to Port Packet Drop and Errors section in the GigaVUE Administration Guide.

The following table describes the arguments for the **threshold** command:

Argument	Description
rx	Configures threshold parameters for rx direction.
tx	Configures threshold parameters for tx direction.
drop	Configures the drop threshold parameters.
error	Configures the error threshold parameters.
global | circuit | hybrid | inline-net | inline-tool | network | stack | tool | gs	Configures threshold parameters as follows: • global - global threshold value (applicable for all port types) • circuit - for circuit ports • hybrid - for hybrid ports • inline-net - for inline network ports • inline-tool - for inline tool ports • network - for network ports • stack - for stack ports • tool - for tool ports • gs - for GigaSMART ports
count <value>	Configures threshold count. Value must be greater than zero.
percent <value>	Configures threshold percent. Allowable range is 0-100. Percentage value can be configured as low as 0.0001%

Related Commands

The following table summarizes other commands related to the **threshold** command:

Task	Command
Deletes the configured threshold values	**no threshold rx/tx drop/error type ***

tool-mirror

Required Command-Line Mode = Configure

Use the **tool-mirror** command to configure a pass-all between two tool ports or a tool port and a tool GigaStream on the same node irrespective of the maps already in place for the ports. Refer to the *"Working with Map-Passalls and Port Mirroring"* section in the *GigaVUE Fabric Management Guide* for a discussion of use cases for tool-mirrors.

Keep in mind the following rules and notes when you configure tool-mirrors:

- You can only use tool-mirror connections between tool ports/GigaStream on the same node. Cross-node tool-mirror connections are not supported.
- Tool-mirrors are not supported on tool ports with copper SFPs installed or on 100Gb ports with CFP2 transceivers.
- In a cluster, tool-mirror is supported only on the leader.

The **tool-mirror** command has the following syntax:

```
tool-mirror <alias <alias>>
  from <port-id | port-alias | port-list | inline-network-alias | inline-network-group-alias>
  to <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias |
     inline-tool-group-alias | inline-serial-alias | bypass> [comment <comment>]
```

The following table describes the arguments for the **tool-mirror** command:

Argument	Description
alias <alias>	Specifies the alias by which this tool-mirror will be known.
from <port-id | port-alias | port-list | inline-network-alias | inline-network-group-alias>	Specifies the source tool port(s) for the tool-mirror. Use one of the following: • **port-id, port-alias, port-list**—Specifies the source tool ports using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **inline-network-alias**—Specifies the source tool ports using the specified inline network alias. • **inline-network-group-alias**—Specifies the source tool ports using the specified inline network group alias.
to <port-id | port-alias | port-list | gigastream-alias | gigastream-alias-list | inline-tool-alias | inline-tool-group-alias | inline-serial-alias | bypass>	Specifies the destination tool port(s) for the tool-mirror. Use one of the following: • **port-id, port-alias, port-list**—Specifies the destination tool ports using the standard conventions described in Port Lists Definition in the GigaVUE-OS. • **gigastream-alias, gigastream-alias-list**—Specifies the destination tool ports using the specified tool GigaStream. Refer to the *"GigaStream"* section in the *GigaVUE Fabric Management*

Argument	Description
	Guide for details on GigaStream. • **inline-tool-alias**—Specifies the destination tool ports using the specified inline tool alias. • **inline-tool-group-alias**—Specifies the destination tool ports using the specified inline tool group alias. • **inline-serial-alias**—Specifies the destination tool ports using the specified inline tool series alias. • **bypass**—Specifies the destination tool ports using the specified inline bypass.
comment <comment>	Specifies a comment in up to 128 characters. Comments longer than one word must be enclosed in double quotation marks.

The following table shows some examples using the **tool-mirror** command:

Command	Comments
(config) # tool-mirror alias toolpass from 1/3/x1 to 1/3/x12	Configures a tool-mirror between tool port 1/3/x1 and tool port 1/3/x12. The tool-mirror has an alias of **toolpass**.
(config) # tool-mirror alias streampass from 14/2/q1 to mygigastream	Configures a tool-mirror from 14/2/q1 to the GigaStream with the alias **mygigastream**. The tool-mirror has an alias of **streampass**.

Related Commands

The following table summarizes other commands related to the **tool-mirror** command:

Task	Command
Displays all tool-mirror connections.	**# show tool-mirror**
Displays a specified tool-mirror connection.	**# show tool-mirror alias Tmirr**
Displays all tool-mirrors.	**# show tool-mirror all**
Displays all tool-mirrors in table format.	**# show tool-mirror brief**
Deletes a specified tool-mirror.	**(config) # no tool-mirror alias Tmirr**
Deletes all tool-mirrors.	**(config) # no tool-mirror all**

traceroute

Required Command-Line Mode = Enable

Use the **traceroute** command to trace the route packets take to a destination.

The **traceroute** command has the following syntax:

```
traceroute [ -46dFITUnrAV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ]
[ -p port ]
            [ -t tos ] [ -l flow_label ] [ -w waittime ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ]
host [ packetlen ]
```

These are standard Linux options for **traceroute**. Refer to online man pages for details.

tunnel

Required Command-Line Mode = Configure

Required License: Advanced Feature License to configure circuit ports on GigaVUE-OS® TA Series nodes

Use the **tunnel** command to configure tunnels to encapsulate and decapsulate traffic. For details about the different types of native tunnels, which are independent of GigaSMART operations, refer to the *"Tunnels"* chapter in the *GigaVUE Fabric Management Guide*.

Refer to the following sections for information about how to configure various types of native tunnels:

- L2-Circuit Tunnel
- Layer 2 Generic Routing Encapsulation (L2GRE) Tunnel
- Virtual Extensible LAN (VXLAN) Tunnel

NOTE: Configuration of different types of Encapsulation and Decapsulation tunnel (L2GRE & VxLAN and vice versa) on the same IP interface is not valid though CLI allows such configuration.

L2-Circuit Tunnel

L2-Circuit tunnel is a type of tunnel that uses circuit-ID to encapsulate the traffic. These tunnels are bidirectional. For details about the L2-Circuit tunnels, refer to the *"About Circuit-ID Tunnels"* section in the *GigaVUE Fabric Management Guide*.

Configure L2-Circuit Tunnel for Encapsulation

Create an L2-Circuit tunnel for encapsulation using the **tunnel** command, which has the following syntax:

```
tunnel alias <alias> encap l2-circuit
  circuit-id <value> => value between 2 to 4000>
```

The following table describes the arguments for the L2-Circuit tunnel for encapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> encap l2-circuit	Specifies an alias for the circuit tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. Also, specify the mode as encap and the type as l2-circuit. For example: **(config) # tunnel alias <alias> encap l2-circuit**
circuit-id <value> => value between 2 to 4000>	Specifies the circuit-ID used to encapsulate the traffic. The valid range is 2–4000. For example: **(config tunnel alias <alias> encap l2-circuit) # circuit-id 2000**

NOTE: Use this encap tunnel in the map configuration to encapsulate the traffic with l2-circuit-id tunnel and send the traffic to the circuit tool ports of the map.

Configure L2-Circuit Tunnel for Decapsulation

Create an L2-Circuit tunnel for decapsulation using the **tunnel** command, which has the following syntax:

```
tunnel alias <alias> decap l2-circuit
  circuit-id <value/range> // Range is value1,value2,value3
  attach <circuit-port-list / circuit-gigastream>
```

The following table describes the arguments for the L2-Circuit tunnel for decapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> decap l2-circuit	Specifies an alias for the circuit tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. Also, specify the mode as decap and the type as l2-circuit. For example: **(config) # tunnel alias <alias> decap l2-circuit**
circuit-id <value/range> // Range is value1,value2,value3	Specifies the circuit-ID used to encapsulate the traffic. The valid range is 2–4000. For example: **(config tunnel alias <alias> decap l2-circuit) # circuit-id 100,200,300**
attach <circuit-port-list /	Specifies the circuit ports or circuit GigaStream to be attached with the

Argument	Description
circuit-gigastream>	circuit tunnel. For example: **(config tunnel alias <alias> decap l2-circuit) # attach <circuit-port-list>**

Layer 2 Generic Routing Encapsulation (L2GRE) Tunnel

L2GRE tunnels are used to route traffic from any remote device to a GigaVUE HC Series or GigaVUE TA Series device over the internet. For details about the L2GRE tunnels, refer to the *"About Layer 2 Generic Routing Encapsulation (L2GRE) Tunnels"* section in the *GigaVUE Fabric Management Guide*.

Configure L2GRE Tunnel for Encapsulation

Create a L2GRE tunnel for encapsulation using the tunnel command, which has the following syntax:

tunnel alias <alias> encap l2gre
comment <description>
attach <ip-interface-name>
ipdst <destination IP address>
exit

The following table describes the arguments for the L2GRE tunnel for encapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> encap l2gre	Specifies an alias for the L2GRE tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # tunnel alias <alias> encap l2gre**
comment <description>	Specifies a description for the L2GRE tunnel. For example: **(config tunnel alias <alias> encap l2gre) # comment <description>**
attach <ip-interface-name>	Specifies the IP interface to which you have attached the circuit port you have configured on the device. For example: **(config tunnel alias <alias> encap l2gre) # attach <ip-interface-name>**
ipdst <destination IP address>	Specifies the IP address of the destination device of the tunnel.

Argument	Description
	For example: **(config tunnel alias <alias> encap l2gre) # ipdst <destination IP address>**
exit	Exits the L2GRE tunnel configuration. For example: **(config tunnel alias <alias> encap l2gre) # exit**

NOTE: Use this encap tunnel in the map configuration to encapsulate the traffic with L2GRE ID and send the traffic to the required destination device, where the traffic will be decapsulated.

Configure L2GRE Tunnel for Decapsulation

Create a L2GRE tunnel for decapsulation using the tunnel command, which has the following syntax:

```
tunnel alias <alias> decap l2gre
  comment <description>
  attach <ip-interface-name>
  exit
```

The following table describes the arguments for the L2GRE tunnel for decapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> decap l2gre	Specifies an alias for the L2GRE tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # tunnel alias <alias> decap l2gre**
comment <description>	Specifies a description for the L2GRE tunnel. For example: **(config tunnel alias <alias> decap l2gre) # comment <description>**
attach <ip-interface-name>	Specifies the IP interface to which you have attached the circuit port you have configured on the device. For example: **(config tunnel alias <alias> decap l2gre) # attach <ip-interface-name>**
exit	Exits the L2GRE tunnel configuration. For example: **(config tunnel alias <alias> decap l2gre) # exit**

NOTE: Use this decap tunnel in the map configuration to decapsulate the traffic with L2GRE tunnel and send the traffic to the required tool ports of the map.

Configure L2GRE Group

Create a L2GRE group using the tunnel command, which has the following syntax:

```
tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>
   add <l2gre-id-list>
  comment <description>
  delete <l2gre-id-list>
  exit
```

The following table describes the arguments for configuring the L2GRE group using the **tunnel** command:

Argument	Description
tunnel l2gre [box-id <id|all>] l2gre-group alias<l2gre-group-name>	Specifies the box identifier in which the L2GRE IDs will be added. Also, specifies an alias for the L2GRE group. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # [no] tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>**
add <l2gre-id-list>	Adds the L2GRE IDs for the device. The range is 1 to 4294967295. For example: **(config tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>) #add <l2gre-id-list>**
comment <description>	Specifies the description for the L2GRE group. For example: **(config tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>) #comment <description>**
delete <l2gre-id-list>	Deletes the specific L2GRE group. For example: **(config tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>) #delete <l2gre-id-list>**
exit	Exits the L2GRE group configuration. For example: **(config tunnel l2gre [box-id <id|all>] l2gre-group alias <l2gre-group-name>) #exit**

Configure L2GRE ID

The L2GRE ID that you have added in the L2GRE group created for a specific device must be configured either at the chassis-level or at the network port-level. The ID that you configure for the network port overrides the ID configured for the chassis. The L2GRE IDs help to identify the respective tunnels.

Configure L2GRE ID for a chassis using the tunnel command, which has the following syntax:

```
tunnel l2gre box-id <id> global-encap-id <l2gre ID>
  exit
```

The following table describes the arguments for configuring the L2GRE ID at the chassis-level using the **tunnel** command:

Argument	Description
tunnel l2gre box-id <id> global-encap-id <l2gre ID>	Specifies the box identifier for which the L2GRE ID will be added. Also, specifies the L2GRE ID that will be configured for the chassis. For example: **(config) # tunnel l2gre box-id <id> global-encap-id <l2gre ID>**
exit	Exits the L2GRE ID configuration. For example: **(config) # tunnel l2gre box-id <id> global-encap-id <l2gre ID> #exit**

NOTE: To configure L2GRE ID for a network port, refer toport.

Virtual Extensible LAN (VXLAN) Tunnel

VXLAN is a simple tunneling mechanism that allows overlaying a Layer 2 (L2) network over a Layer 3 (L3) underlay with the use of any IP routing protocol. It uses MAC Address-in-User Datagram Protocol (MAC-in-UDP) encapsulation. For details about the VXLAN tunnels, refer to the *"About Virtual Extensible LAN (VXLAN) Tunnels"* section in the *GigaVUE Fabric Management Guide*.

Configure VXLAN Tunnel for Encapsulation

Create a VXLAN tunnel for encapsulation using the tunnel command, which has the following syntax:

```
tunnel alias <alias> encap vxlan
  comment <description>
  attach <ip-interface-name>
  ipdst <destination IP address>
```

l4srcport <layer4 source port number>

exit

The following table describes the arguments for the VXLAN tunnel for encapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> encap vxlan	Specifies an alias for the VXLAN tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # tunnel alias <alias> encap vxlan**
comment <description>	Specifies a description for the vxlan tunnel. For example: **(config) tunnel alias <alias> encap vxlan) # comment <description>**
attach <ip-interface-name>	Specifies the IP interface to which you have attached the circuit port you have configured on the device. For example: **(config tunnel alias <alias> encap vxlan) # attach <ip-interface-name>**
ipdst <destination IP address>	Specifies the IP address of the destination device of the tunnel. For example: **(config tunnel alias <alias> encap l2gre) # ipdst <destination IP address>**
l4srcport <layer4 source port number>	Specifies the layer 4 source port number For example: **(config tunnel alias <alias> encap vxlan) # l4srcport <layer4 source port number>**
exit	Exits the VXLAN tunnel configuration. For example: **(config tunnel alias <alias> encap vxlan) # exit**

NOTE: Use this encap tunnel in the map configuration to encapsulate the traffic with VXLAN ID and send the traffic to the required destination device, where the traffic will be decapsulated.

Configure VXLAN Tunnel for Decapsulation

Create a VXLAN tunnel for decapsulation using the tunnel command, which has the following syntax:

```
tunnel alias <alias> decap vxlan
   comment <description>
  attach <ip-interface-name>
  exit
```

The following table describes the arguments for the VXLAN tunnel for decapsulation using the **tunnel** command:

Argument	Description
tunnel alias <alias> decap vxlan	Specifies an alias for the VXLAN tunnel. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # tunnel alias <alias> decap vxlan**
comment <description>	Specifies a description for the VXLAN tunnel. For example: **(config tunnel alias <alias> decap vxlan) # comment <description>**
attach <ip-interface-name>	Specifies the IP interface to which you have attached the circuit port you have configured on the device. For example: **(config tunnel alias <alias> decap vxlan) # attach <ip-interface-name>**
exit	Exits the VXLAN tunnel configuration. For example: **(config tunnel alias <alias> decap vxlan) # exit**

NOTE: Use this decap tunnel in the map configuration to decapsulate the traffic with VXLAN tunnel and send the traffic to the required tool ports of the map.

Configure VXLAN Group

Create a VXLAN group using the tunnel command, which has the following syntax:

```
tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>
   add <vxlan-id-list>
  comment <description>
  delete <vxlan-id-list>
  exit
```

The following table describes the arguments for configuring the VXLAN group using the **tunnel** command:

Argument	Description
tunnel vxlan [box-id	Specifies the box identifier in which the VXLAN IDs will be added. Also,

Argument	Description
<id|all>] vxlan-group alias<vxlan-group-name>	specifies an alias for the VXLAN group. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # [no] tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>**
add <vxlan-id-list>	Adds the VXLAN IDs for the device. The range is 1 to 4294967295. For example: **(config tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>) #add <vxlan-id-list>**
comment <description>	Specifies the description for the VXLAN group. For example: **(config tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>) #comment <description>**
delete <vxlan-id-list>	Deletes the specific VXLAN group. For example: **(config tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>) #delete <vxlan-id-list>**
exit	Exits the VXLAN group configuration. For example: **(config tunnel vxlan [box-id <id|all>] vxlan-group alias <vxlan-group-name>) #exit**

Configure VXLAN ID

The VXLAN ID that you have added in the VXLAN group created for a specific device must be configured either at the chassis-level or at the network port-level. The ID that you configure for the network port overrides the ID configured for the chassis. The VXLAN IDs help to identify the respective tunnels.

Configure VXLAN ID for a chassis using the tunnel command, which has the following syntax:

```
tunnel vxlan box-id <id> global-encap-id <vxlan ID>
  exit
```

The following table describes the arguments for configuring the VXLAN ID at the chassis-level using the **tunnel** command:

Argument	Description
tunnel vxlan box-id <id>	Specifies the box identifier for which the VXLAN ID will be added. Also,

Argument	Description
global-encap-id <vxlan ID>	specifies the VXLAN ID that will be configured for the chassis. For example: **(config) # tunnel vxlan box-id <id> global-encap-id <vxlan ID>**
exit	Exits the VXLAN ID configuration. For example: **(config) # tunnel vxlan box-id <id> global-encap-id <l2gre ID> #exit**

NOTE: To configure VXLAN ID for a network port, refer toport.

The following table summarizes other commands related to the **tunnel** command:

Task	Command
Displays all circuit tunnels.	**(config) # show tunnel**
Displays the specific tunnel alias.	(config) # show tunnel <alias>
Displays statistics specific to the given tunnel alias.	(config) # show tunnel stats <alias>
Displays the statistics data of all the encapsulation tunnels configured for the device.	(config) # show tunnel stats all
Displays the L2GRE ID statistics data.	(config) # show tunnel l2gre [box-id <box-id | all>] [l2gre-group alias <group-name> | all]
Displays the VXLAN ID statistics data.	(config) # show tunnel vxlan [box-id <box-id | all>] [alias <group-name> | all]
Displays the L2GRE ID configured for the chassis.	(config) # show tunnel l2gre global-encap-id
Displays the VXLAN ID configured for the chassis.	(config) # show tunnel vxlan global-encap-id
Clears all the L2GRE statistics data.	(config) # clear tunnel l2gre l2gre-group stats all
Clears all the VXLAN statistics data.	(config) # clear tunnel vxlan vxlan-group stats all
Clears the statistics data of the L2GRE tunnels.	(config) # clear tunnel l2gre stats all
Clears the statistics data of the VXLAN tunnels.	(config) # clear tunnel vxlan stats all
Clears the statistics data for the specified encapsulation tunnel.	(config) # clear tunnel stats alias <tunnel alias>
Clears the statistics data for all the encapsulation tunnels configured for the device.	(config) # clear tunnel stats all

Task	Command
Deletes a specified circuit tunnel by the alias. If the circuit tunnel is associated with a map, it cannot be deleted.	**(config) # no tunnel alias <alias>**
Deletes all circuit tunnels. If the circuit tunnel is associated with a map, it cannot be deleted.	**(config) # no tunnel all**
Removes the L2GRE ID that is configured for the chassis.	**(config) # no tunnel l2gre box-id <box-id> global-encap-id**
Removes the VXLAN ID that is configured for the chassis.	**(config) # no tunnel vxlan box-id <box-id> global-encap-id**

tunnel-endpoint

Required Command-Line Mode = Configure

Use the **tunnel-endpoint** command to configure a tunnel endpoint that is a destination for traffic from a L2GRE tunnel. Using stateless or stateful load balancing, GigaSMART can be configured to distribute the traffic from a tunnel to multiple tunnel endpoints.

There is no mapping of a tunnel endpoint to a GigaSMART group (gsgroup). A tunnel endpoint is only mapped when a GigaSMART operation (gsop) is configured for **tunnel-encap type l2gre**. Refer to gsop for details.

For information on L2GRE encapsulation, refer to the *"GigaSMART Layer 2 GRE Tunnel Encapsulation/Decapsulation" section in the GigaVUE Fabric Management Guide*.

The **tunnel-endpoint** command has the following syntax:

```
tunnel-endpoint te-id <tunnel endpoint ID>
  alias <alias>
  type remote ip-address <IP address>
```

The following table describes the arguments for the **tunnel-endpoint** command:

Argument	Description
tunnel-endpoint te-id <tunnel endpoint ID>	Specifies the identifier of the tunnel endpoint. The tunnel endpoint ID has the format teN, where N is a value from 1 to 128, for example: te1 to te128. A maximum of 128 tunnel endpoints can be configured.
alias <alias>	Specifies an optional alias for the tunnel endpoint. The alias must be unique and can contain up to 128 characters. Aliases are case-sensitive. For example: **(config) # tunnel-endpoint te-id te1 alias Tunnel-endpoint1**

Argument	Description
	NOTE: The alias cannot use the same format as the tunnel endpoint identifier, te1 to te128.
type remote ip-address <IP address>	Specifies the type, which is remote, and specifies the IP address of the tunnel endpoint. Only one IP address can be configured for each tunnel endpoint. The destinations must be reachable from the tunnel IP. For example: **(config) # tunnel-endpoint te-id te1 type remote ip-address 1.1.1.1** **NOTE:** Once a tunnel endpoint is associated with a port group, the IP address cannot be changed.

The following table summarizes other commands related to the **tunnel-endpoint** command:

Task	Command
Displays all tunnel endpoints.	**# show tunnel-endpoint**
Displays a specified tunnel endpoint by the alias.	**# show tunnel-endpoint alias Tunnel-endpoint1**
Displays tunnel endpoints status.	**# show ip destination gsgroup <gsgroup-alias>**
Displays tunnel endpoints statistics.	**# show ip destination stats gsgroup <gsgroup-alias>**
Displays a specified tunnel endpoint by the tunnel endpoint identifier.	**# show tunnel-endpoint te-id te1**
Displays all port groups and displays the associated tunnel endpoint.	**# show port-group all**
Displays load balancing statistics for a specified port group, including stats for tunnel endpoint.	**# show load-balance port-group stats alias portgrp1**
Displays load balancing statistics for all port groups, including stats for tunnel endpoint.	**# show load-balance port-group stats all**
Deletes a specified tunnel endpoint by the alias.	**(config) # no tunnel-endpoint alias Tunnel-endpoint1**
Deletes all tunnel endpoints. If the tunnel endpoint has been associated with a port group, it cannot be deleted.	**(config) # no tunnel-endpoint all**
Deletes a specified tunnel endpoint by its tunnel endpoint identifier. If the tunnel endpoint has been associated with a port group, it cannot be deleted.	**(config) # no tunnel-endpoint te-id te1**

uboot

Required Command-Line Mode = Configure

Required User Level = Admin

Use the **uboot install** command to install the binary bootloader code included with the active/booted image.

username

Required Command-Line Mode = Configure

Use the **username** command to manage local user accounts on GigaVUE-OS nodes. You can configure different user account levels—**admin** and **monitor**—so that each user has rights that are appropriate for the type of work they will be doing with the system. You can also remove user accounts (or parts of their configuration) with the **no username** command.

The **username** command has the following syntax:

```
username <username>
  disable [login | password]
 full-name <full name>
password <prompt | cleartext password>
 roles <add <user role> [user role] | replace <user role> [user role]>
```

The following table describes the arguments for the **username** command:

Argument	Description
username <username>	Creates a name for a user account. The system comes initially with two accounts already created, as follows: • **admin** users have access to the full range of features and functionality available on the system. They can configure packet distribution, configure users, view logs, and so on—if it can be done on the GigaVUE-OS node, an **admin** user can do it. • **monitor** users do not have access to any port configuration settings. Their access consists mainly of the ability to use the **show** command to see what basic settings are in place on the node. **NOTES:** • New users are created with the Default role automatically. **default** users have access to all command modes. However, they do not have access to unassigned ports. • Remote usernames that include a forward slash (/) are not supported in GigaVUE-OS. A remote username is one created in a remote server. The workaround is to use the backward slash (\) in remote usernames.
disable[monitor]	For security purposes, monitor account must be disabled by default. To enable the monitor account, you must specifically enable the account and create a

Argument	Description
	password. **(config) # no username <username> monitor disable** **(config) # username <username> monitor password xxxxx** **NOTE:** An encrypted version of password will be displayed to the user.
disable [login | password]	**login** Temporarily disables logins for the specified account. Disabling an account closes any currently open sessions for the specified account. To reverse a disabled account, use the following: **(config) # no username <username> disable** Use the following command to lock out access to a user account: **(config) # username <username> disable login** **NOTE:** You cannot disable the **admin** account. **password** This command is used to disable login for the specified local account. Remote accounts that are mapped to this disabled local account will continue to work. Use the following command to disable the local login for the user account: **(config) # username <username> disable password**
full-name <full name>	Specifies the full name for the account (sometimes referred to as the **gecos**). The full name string may contain spaces and other characters, but must be contained in quotation marks. For example: **(config) # username John full-name "IT User"** The full name appears in the CLI output of the **show usernames** command.
password <prompt | cleartext password>	Adds or changes the password for the specified user account. Refer to Password Policies for minimum password requirements.
<add <user role> [user role] | replace <user role> [user role]>	Adds to or replaces roles from the specified user account. The roles themselves are configured with the **aaa authorization roles** command. Refer to aaa for details. Roles provide users with different levels of access to ports with the same role assigned. Refer to the "Configuring Series Security Options" in the *GigaVUE Administration Guide* for details.

The following table summarizes other commands related to the **username** command:

Task	Command
Displays user names and account status.	**# show usernames**
Displays the currently logged in users.	**# show users**
Displays a history of user logins.	**# show users history**
Displays a history of user logins for a specified username.	**# show users history username monitor**
Displays the roles assigned to the logged in users.	**# show users roles**
Displays a specified user name and assignment.	**# show usernames assignment alias admin**
Displays all specified user names and assignment.	**# show usernames assignment all**
Deletes a specified user account.	**(config) # no username operator**
Re-enables a specified user account.	**(config) # no username monitor disable**
Re-enables login for a specified user account.	**(config) # no username monitor disable login**
Re-enables login password for a specified user account.	**(config) # no username monitor disable password**
Deletes the full name of the specified user.	**(config) # no username monitor full-name**
Deletes a specified authorization role from a specified user account.	**(config) # no username monitor roles add admin**
Deletes all authorization roles from a specified user account.	**(config) # no username monitor roles all**

Access for Read-Only Users

To give read-only access to a monitor user to a certain set of ports/maps, add the system built-in role, monitor, together with another role that has port/maps assigned.

For example, with the admin user, create a new role called NetOpsRole with read/write access to ten maps. Create a monitor user with a username of NetmonUser. Assign both roles of monitor and NetOpsRole to username NetmonUser. This will give NetmonUser read-only access to the ten maps.

For example, the following **username** commands create a new **admin** user and a new **monitor** user:

Command	Comments
(config) # username psandoval password Nine9.Eight8! **(config) # username psandoval roles add admin**	Creates a new account named **psandoval** with a password and grants it admin privileges.
(config) # username bcrawford password Seven7.Six6!	Creates a new account named **bcrawford** with a password. New users are automatically created with default operator level privileges, so there is no need to grant an additional role.
(config) # username aaron password Five5.Four4! **(config) # username aaron roles add monitor**	Creates a new account named **aaron** with a password and grants it monitor privileges.

Once you have configured user accounts, use the **show usernames** command to review your settings. Access for Read-Only UsersThe following figure shows the output after a few users have been added.

```
gigamon-28013a (config) # sh username
USERNAME      FULL NAME                 ACCOUNT STATUS
=============================================================================
admin         System Administrator      Password set
anthony                                 Local password login disabled
mScutaro                                Password set
monitor       System Monitor            Account locked out
operator      System Operator           Account locked out
psandoval                               Account disabled
```

NOTE: If the logged in user has a **monitor** role, the account status of other users will not be displayed in the output of the **show usernames** command as shown in the following figure.

```
(config) # show usernames
USERNAME      FULL NAME                 ACCOUNT STATUS
aaron                                   (not available)
admin         System Administrator      (not available)
anthony                                 (not available)
```

Change Passwords

The default password on the **admin** account must be changed to a non-default value.

If you log into GigaVUE-OS with the default **admin** password, the **configuration jump-start** automatically starts and forces a password change as follows:

login: adminGigamon GigaVUE-OSGigaVUE-OS configuration wizardStep 1: Hostname? [gigamon] MyNodeStep 2: Management interface? [eth0]Step 3: Use DHCP on eth0 interface? noStep 4: Use zeroconf on eth0 interface? [no]Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.10.10.10/24Step 6: Default gateway? 10.10.10.1Step 7: Primary DNS server? 10.10.1.20Step 8: Domain name? Step 9: Enable IPv6? [yes]Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]Step 11: Enable DHCPv6 on eth0 interface? [no]Step 12: Enable secure cryptography? [no]Step 13: Enable secure passwords? [no]Step 14: Minimum password length? [8]Step 15: Admin password?Please enter a password. Password is a must.Step 15: Admin password?Step 15: Confirm admin password?

For error messages associated with changing the default password on the **admin** account, refer to the *"Changing the Password on admin Account"* section in the *GigaVUE Administration Guide* for details.

Password Policies

Internal Change Notes

GigaVUE-OS nodes observe several policies designed to ensure strong password protection for user accounts.

Policy	Description
Password Standards	Passwords must meet the following standards: • include 8-64 characters • include at least one numeral • include at least one lower case letter • include at least one upper case letter • include at least one special character (for example, **!**, **#**, **$**, **%**, **^**, **&**, or ***** – ASCII 0x21, 0x2F, 0x3A, 0x40, 0x5B, 0x5F, 0x7B, 0x7E) • include at least eight characters must differ from the old password and the new password **Note:** The following special characters are not supported: • **"** • **/**

Policy	Description
	• ? However, you can use these characters in the password as described in the Reserved Characters in Passwordssection. • must not include user-name or parts of full-name Refer to System for details. Refer to Configure Secure Password for details.
Password Recommendations	The following are password recommendations: • passwords should be configured on all user accounts • passwords should be changed on default accounts such as monitor accounts • passwords should be unique, meaning never used elsewhere or at another time • passwords should not be shared, meaning each user account should have their own password • passwords should be long, meaning at least 15 to 20 characters • passwords should be complex, meaning a mix of numerals, upper case letters, lower case letters, and special characters **NOTE:** It is recommended that you do not include the at sign (@) in passwords. Under some circumstances, this can lead to the failure of some CLI commands, such as **image fetch** or **configuration upload**. **NOTE:** The monitor account is designed to give a read-only access to the GigaVUE-OS. The monitor account is disabled by default. To enable it, assign a password to the account. GigaVUE-FM, GigaVUE-OS and GigaVUE-OS CLI users can use the monitor account as long as it is enabled (has a

Policy	Description
	password).
Password recommendation for admin users	The default password on the admin account is admin123A!. After the first login, you must change the password to a non-default value in compliance with the password requirements mentioned above. For example: gigamon123A!!.
Password Change Rights	Only **admin** users can change the passwords of other users. For example, to change the password of the **psandoval** account, an **admin** user would use the following command: **(config) # username psandoval password <new password>**

A secure passwords mode is available. Refer to the *"Configuring Secure Passwords Mode"* section in the *GigaVUE Administration Guide* for details, as well as thesystem.

Reset Passwords on GigaVUE-HC1, and GigaVUE-HC3

To reset a password on GigaVUE-HC1, and GigaVUE-HC3 nodes use the following steps:

1. Reboot the system.
2. Watch the screen messages until you see the following:

   ```
   Press Enter to boot this image, or any other key for boot menu
   ```
3. Press any key other than Enter, such as the space bar
4. To modify the kernel arguments, type: a

   This modifies the kernel arguments before booting. The kernel arguments are:

   ```
   <112M memmap=2176M$1920M console=tty0 console=ttyS0,115200n8
   ```
5. Add the following to the kernel arguments: reset_button=1, as follows

   ```
   <112M memmap=2176M$1920M console=tty0 console=ttyS0,115200n8 reset_button=1
   ```

This performs a reboot and causes the admin password to be reset to a blank password, and it also reconfigures AAA authentication to use local authentication first.

6. When the system boots up, login as the **admin** user and set a new admin password using the following command: **username admin password <new password>**.

Reset Passwords on GigaVUE-HCT, and GigaVUE-HC1-Plus

To reset a password on GigaVUE-HCT, and GigaVUE-HC1-Plus nodes use the following steps:

1. Reboot the system.
2. Watch the screen messages until you see the following, and press the **"ESC"** key several times to stop the bootup process:

```
HROT-BL -"6.0.00" (release/hd_6000,dc1efdffdb7)  Aug 20 2022:17:11:34
[NOTICE] - HROT-BL Booting Firmware

HROT-PFW -"6.0.00" (release/hd_6000,74671ff592a) Aug 28 2022:13:49:17
[NOTICE] - HROT-PFW Primary BIOS Integrity: [ok]
[NOTICE] - HROT-PFW CN:ENG-ISK1 O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW CN:ENG-ISK2 O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW CN:PROD-ISK1 O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW CN:ENG-KEK-PlainSigner O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW CN:ENG-KEK2-PlainSigner O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW CN:PK O:Gigamon Integrity: [ok]
[NOTICE] - HROT-PFW Primary BIOS and KEY authentication successful
Booting.Yantra.40.011p.0  06/21/2022  12:00

 POST CODE: 0x000000C2
 POST CODE: 0x000000C5
 POST CODE: 0x000000C6
 POST CODE: 0x000000C3
 POST CODE: 0x000000C7
 POST CODE: 0x000000C8
 POST CODE: 0x0000007E
 POST CODE: 0x00000083
 POST CODE: 0x000000C1
 POST CODE: 0x000000C4
 POST CODE: 0x000000A2
 POST CODE: 0x0000004B
 POST CODE: 0x00000052
 POST CODE: 0x0000004D
 POST CODE: 0x000000A2
 POST CODE: 0x00000059
 POST CODE: 0x00000010
 POST CODE: 0x00000011
 POST CODE: 0x00000012
 POST CODE: 0x00000013
 POST CODE: 0x00000015
 POST CODE: 0x00000016

 POST CODE: 0x00000017
 POST CODE: 0x00000018
 POST CODE: 0x0000001D

 POST CODE: 0x00000026
 POST CODE: 0x00000016
 POST CODE: 0x00000017
 POST CODE: 0x00000018
```

```
POST CODE: 0x00000020
POST CODE: 0x00000016
POST CODE: 0x00000017
POST CODE: 0x00000018
POST CODE: 0x00000027
POST CODE: 0x00000028
POST CODE: 0x0000002B
POST CODE: 0x00000029
POST CODE: 0x0000002E
POST CODE: 0x00000030
POST CODE: 0x000000FB
```

3. Go to the GRUB prompt, press the key "E" when the image option appears in the display. The grub prompt appears as follows:

```
setparams 'GigaVUE-OS-image1' 'GigaVUE-OS1'
set gfxpayload=text
insmod gzio
insmod part_msdos
insmod xfs
set root='hd0,gpt2'
linuxefi (hd0,2)/vmlinuz console=ttyS0,115200n8 text img_id=1 root=/de\
v/sda5 rw net.ifnames=0 biosdevname=0 acpi=copy_dsdt pcie_aspm=force resume=/d\
ev/sda4 quiet
```

4. Add reset_button=1 at the end as below and then press **"ctrl + x"**

```
linuxefi (hd0,2)/vmlinuz console=ttyS0,115200n8 text img_id=1 root=/de\ v/sda5
rw net.ifnames=0 biosdevname=0 acpi=copy_dsdt pcie_aspm=force resume=/d\
ev/sda4 quiet reset_button=1
```

Reset Passwords on GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-TA200, GigaVUE-TA200E, GigaVUE-TA400, and GigaVUE-TA400E

To reset a password on GigaVUE-TA25, GigaVUE-TA25E, GigaVUE-TA200, GigaVUE-TA200E, GigaVUE-TA400, GigaVUE-TA400E nodes use the following steps:

1. Reboot the system.
2. Go to the GRUB prompt, press the key "E" when the image option appears in the display. The grub prompt appears as follows:

```
setparams 'GigaVUE-OS-image1' 'GigaVUE-OS1'
set gfxpayload=text
insmod gzio
insmod part_msdos
insmod xfs
set root='hd0,gpt2'
linuxefi (hd0,2)/vmlinuz console=ttyS0,115200n8 text img_id=1 root=/de\
v/sda5 rw net.ifnames=0 biosdevname=0 acpi=copy_dsdt pcie_aspm=force resume=/d\
```

```
ev/sda4 quiet
```

3. Add reset_button=1 at the end as below and then press **"ctrl + x"**

```
linuxefi (hd0,2)/vmlinuz console=ttyS0,115200n8 text img_id=1 root=/de\ v/sda5
rw net.ifnames=0 biosdevname=0 acpi=copy_dsdt pcie_aspm=force resume=/d\
ev/sda4 quiet reset_button=1
```

Configure a Password Expiration Duration

Use the following CLI command to configure the number of days before a password expires:

(config) # aaa authentication password expiration duration 20

Refer to aaa authentication for details.

Configure Login Attempts

Use the following CLI command to configure the handling of failed login attempts:

(config) # aaa authentication attempts

Refer to aaa authentication for details.

Reserved Characters in Passwords

This section describes how to use the following reserved characters in passwords:

- **?**
- ****
- **"**

There are two ways to include these characters in a password:

1. Enter the username without specifying the password

In this technique, you issue the **username** command and include the password argument, but do not actually specify the password. This causes the system to prompt you for the password, allowing you to enter reserved characters directly. For example:

(config) # username mcabrera passwordPassword: *********Confirm: *******

In this example, you could enter a password using a reserved character as follows—for example, **Test123?**

2. Include the escape character before each reserved character

Alternatively, you can include reserved characters in a password specified in the **username** command by using the following:

- Enclose the entire password in double-quotation marks. In particular, use this technique to include the question mark (?) in a password.

- Include the escape character, which is the slash (\), before the single quote (") character or before the slash (\) in a password.

The following table shows some sample passwords:

Command	Password Created
username user1 password "Test123?"	Test123?
username user2 password Test123\"	Test123"
username user3 password Test123	Test123\

vport

Use the **vport** command to configure a GigaSMART virtual port used as an aggregation point for traffic directed to second level maps. Second level maps include an Adaptive Packet Filtering component (gsrule) or a GTP rule (flow-rule).

This command does not apply to GigaVUE TA Seriesnodes.

The **vport** command has the following syntax:

```
vport alias <alias>
  gsgroup <GigaSMART group alias>
  failover-action <vport-bypass | vport-drop | network-bypass | network-drop | network-port-forced-down>
  mode gtp-overlap
```

If a gsgroup has not been assigned to a vport yet, the **vport** command has the following syntax:

```
vport alias <alias>
  gsgroup <GigaSMART group alias>
```

Refer to the *"GigaVUE-OS CLI—Configuration Examples"* chapter in this guide for examples.

The following table describes the arguments for the **vport** command:

Argument	Description
vport alias <alias>	Specifies the alias of the virtual port.
gsgroup <GigaSMART group alias>	Specifies the GigaSMART group associated with the virtual port. For example:

Argument	Description
	(config) # vport alias vport1 gsgroup gsgrp5
failover-action <vport-bypass | vport-drop | network-bypass | network-drop | network-port-forced-down>	Specifies a failover action for the virtual port for SSL Decryption for inline tools as follows: • **vport-bypass**—Specifies that the traffic that was directed to the vport goes via the bypass path. • **vport-drop**—Specifies that the traffic that was directed to the vport is dropped. • **network-bypass**—Specifies that all traffic coming to the inline network is directed via the bypass path. • **network-drop**—Specifies that all traffic coming to the inline network is dropped. • **network-port-forced-down**—Specifies that the inline network ports of the inline network are forced to a down state. The default is **vport-bypass**. For example: **(config) # vport alias vport1 failover-action vport-bypass** **Note**: To configure the failover-action, first assign a gsgroup to the vport.
mode gtp-overlap	Specifies the GTP overlap mode. This is an optional mode to use with GTP forward listing and GTP flow sampling when multiple copies of a GTP packet need to be sent to more than one tool. For example: **(config) # vport alias vport1 mode gtp-overlap** • To configure the mode, first assign a gsgroup to the vport. **Note**: • To configure the mode, first assign a gsgroup to the vport. • When the vport mode is changed from overlap to non-overlap mode, indeterministic behavior will be in the traffic. Hence, it is recommended to do a cluster or chassis reload for the configuration to take Refer to the "*GigaSMART GTP Whitelisting and GTP Flow Sampling*" section in the *GigaVUE Fabric Management Guide* for details.

Related Commands

The following table summarizes other commands related to the **vport** command:

Task	Command
Configures a vport on a specified GigaSMART group.	**(config) # vport alias vport1 gsgroup gsg1**
Displays a specified vport. **Note**: The existing 'show vport' command is enhanced to display the 'SVT mode' for the respective inline-ssl app configured throughGigaVUE-FM.	**# show vport alias vport1**
Displays all vports.	**# show vport all**
Displays statistics for a specified vport.	**# show vport stats alias vport1**
Displays statistics for all vports. **Note**: When a vport is modified in the first level map and the second level map, only the statistics of the new vport is visible. The statistics of the previous vport cannot be retrieved.	**# show vport stats all**
Deletes a specified vport.	**(config) # no vport alias vport1**
Clears the failover action on a specified vport.	**(config) # no vport alias vport1 failover-action**
Deletes all vports.	**(config) # no vport all**

web

Required Command-Line Mode = Configure

Use the **web** command and its arguments to enable and configure the GigaVUE HC Series node's onboard Web server used for GigaVUE-FM access to the node. GigaVUE-FM is Gigamon's Web-based GUI for the GigaVUE HC Series node, providing graphical user interface configuration.

The **web** command has the following syntax:

```
web
  auto-logout <number of minutes>
  client
    ca-list <none | default-ca-list>
    cert-verify
  enable
  http
     enable
    port <port number>
    redirect
  httpd listen
    enable
```

```
    interface <interface>
  https
    certificate
      default-cert
     name <cert-name | system-self-signed>
     regenerate
   enable
   port <port number>
   require-dod-cert
logs <access | error> upload <current | log file number> <upload URL>
proxy
   auth
     authtype <none | basic>
     basic <password <password>> | <username <username>>
   host <IPv4 or IPv6 address> [port <port number>]
server ssl min-version <tls1 | tls1.1 | tls1.2>
session
    auto-logout <number of minutes>
   renewal <number of minutes>
```

The following table describes the arguments for the **web** command.

Argument	Description
auto-logout <number of minutes>	Specifies the maximum duration of user inactivity before a Web session is logged out automatically. For example: **(config) # web auto-logout 30** The minimum value is one (1) minute.
client **ca-list <none | default-ca-list>** **cert-verify**	Specifies Web client settings as follows: • **ca-list none**—Specifies that supplemental certificates are not used. • **ca-list default-ca-list**—Specifies that any certificates added to the default supplemental CA list will be used by default. • **cert-verify**—Turns off the verification of server certificates for HTTPS file transfers. (Verification is on by default.)
enable	Enables the availability of the Web-based GigaVUE-FM GUI for GigaVUE-OS nodes. For example: **(config) # web enable**
http **enable** **port <port number>** **redirect**	Configures HTTP access to the Web-based GigaVUE-FM GUI with the following settings: • **enable**—Enables the availability of HTTP for Web access to GigaVUE-FM. If this is disabled, only HTTPS connections will be accepted (and only then if the HTTPS argument is turned on). • **port**—Specifies the port to be used for HTTP access. The default is port 80. • **redirect**—Specifies whether incoming HTTP connections to GigaVUE-FM should be redirected to the secure HTTP port (either the custom port specified with the **https** argument or the default HTTPS port of 443 if none is specified).

Argument	Description
httpd listen enable interface <interface>	Enables or disables the use of a restricted list of interfaces on which the Web server will accept connections: • If the **enable** option is turned on and at least one statically-configured interface is specified in the list created with the **interface** argument, HTTP and HTTPS connections are only accepted on those specified interfaces. • If the **enable** option is turned off, HTTP and HTTPS requests are accepted on any interface.
https certificate default-cert name <cert-name | system-self-signed> regenerate enable port <port number> require-dod-cert	Configures HTTPS access to the Web-based management console, including the following settings: • **certificate default-cert**—Configures the Web server to use the specified certificate from the certificate database for HTTPS communications. This is the default. • **certificate name**—Specifies a named certificate already defined in the database with a private key already configured. • **certificate system-self-signed**—Specifies the system-self-signed certificate, which is automatically generated. • **certificate regenerate**—Regenerates the system-self-signed certificate used with HTTPS communications. • **enable**—Enables the use of HTTPS for access to GigaVUE-FM. This setting does not turn on the Web server generally, but allows HTTPS connections. • **port**—Specifies the TCP port number to be used for GigaVUE-FM connections using HTTPS. The default is 443. • **require-dod-cert**—Configures the Web server to only accept certificates from a Department of Defense (DoD) authorized certificate authority. The default is disabled.
logs <access | error> upload <current | log file number> <upload URL>	Configures information to upload the following types of Web log files to a remote host from a GigaVUE-OS node: • **access**—Specifies a Web access log file to upload to a remote host. Specify either the keyword **current** or a log file number, followed by the upload URL. • **error**—Specifies a Web error log file to upload to a remote host. Specify either the keyword **current** or a log file number, followed by the upload URL. The **current** log file is not compressed. The numbered log files are compressed. One **current** log file and up to eight access and error log files are archived. Numbered log files are named as follows: • web_access_log.1.gz to web_access_log.8.gz • web_error_log.1.gz to web_error_log.8.gz Use one of the following formats for uploading: FTP, TFTP, SCP, or SFTP. Examples: **(config) # web logs error upload current scp://user1:mypw@1.1.1.1:/home/temp/logfilecurrent.txt** **(config) # web logs access upload 2 ftp://myuser:mypass@192.168.1.1/ftp/logfile2.txt**

Argument	Description
proxy auth authtype <none | basic> basic <password <password>> | <username <username>> host <IPv4 or IPv6 address> [port <port number>]	Configures Web proxy settings to be used for HTTP or FTP downloads. First, set a proxy to be used with the **web proxy host <IPv4 or IPv6 address>** command. If you do not specify a port, the default is 1080. Once you have configured a proxy, use the **auth authtype** argument to specify whether a username and password is required to log in to the proxy (**basic**) or not (**none**). If **web proxy auth authtype** is set to **basic**, configure the actual username and password to use with the **web proxy auth basic username** and **web proxy auth basic password** commands.
server ssl min-version <tls1 | tls1.1 | tls1.2>	Specifies a minimum TLS version for the Web server. The following can be specified: • **tls1**—Specifies TLS1.0 (or higher). This is the lowest TLS version and is the default. • **tls1.1**—Specifies TLS1.1 (or higher). • **tls1.2**—Specifies TLS1.2. This is the highest TLS version. For example: **(config) # web server ssl min-version tls1.2**
session auto-logout <number of minutes> renewal <number of minutes>	Configures session settings: • **auto-logout**—Specifies the maximum lifetime of a Web session cookie. • **renewal**—Specifies the length of time before a session expires that the Web server will issue a new cookie and renew the session. This should be set at least as long as the auto-logout setting so that sessions do not expire before they have a chance to be renewed. For example, with an auto-logout of 20 minutes and a renewal setting of 5 minutes, a session will be renewed 15 minutes (20-5) after it starts. The following error message is displayed if the session renewal is greater than the auto-logout: **(config) # web session auto-logout 12Session renewal threshold must be at least 5 sec less than session auto-logout. Resetting it to 11 min 55 sec.**

Related Commands

The following table summarizes other commands related to the **web** command:

Task	Command
Displays Web-based management console configuration settings and status.	**# show web**
Disables auto-logout, so users are not automatically logged out due to inactivity.	**(config) # no web auto-logout**

Task	Command
Deletes supplemental CA certificates from the HTTPS client.	**(config) # no web client ca-list**
Disables verification of server certificates during HTTPS file transfers.	**(config) # no web client cert-verify**
Disables the availability of the Web-based GigaVUE-FM GUI for GigaVUE-OS nodes.	**(config) # no web enable**
Disables HTTP access to the Web-based management console.	**(config) # no web http enable**
Resets the HTTP port to the default port number (80).	**(config) # no web http port**
Disables redirection to HTTPS.	**(config) # no web http redirect**
Disables Web interface restrictions on access to this system.	**(config) # no web httpd listen enable**
Deletes the specified interface from the Web server access restriction list.	**(config) # no web httpd listen interface eth1**
Deletes a specified Web server certificate from use and revert to the certificate configured with the **crypto certificate default-cert** command.	**(config) # no web https certificate name <cert-name>**
Disables HTTPS access to the Web-based management console.	**(config) # no web https enable**
Resets the HTTPS port to the default port number (443).	**(config) # no web https port**
Allows certificates issued from any authority.	**(config) # no web https require-dod-cert**
Disables Web proxy.	**(config) # no web proxy**
Resets the Web proxy authentication type to the default (none).	**(config) # no web proxy auth authtype**
Clears the password.	**(config) # no web proxy auth basic password**
Clears the username.	**(config) # no web proxy auth basic username**
Resets the time so that a Web session never expires.	**(config) # no web session auto-logout**
Resets the time to wait to renew a session before it expires.	**(config) # no web session renewal**

write

Required Command-Line Mode = Configure

Use the **write** command to save changes to the running configuration as well as to display the commands necessary to recreate the current running configuration.

NOTE: The **write** command provides similar functionality to the **configuration write** command. Refer to configuration. Also refer to Using the configuration Command for information on working with configuration files.

The **write** command has the following syntax:

```
write
  memory [local]
  terminal
```

The following table describes the arguments for the **write** command:

Command	Description
memory [local]	Saves the running configuration to a active configuration file. For example: **(config) # write memory** In a cluster environment: • **write memory** on a leader is propagated to all the nodes in the cluster, and the configurations are saved locally for each of the nodes. • **write memory local** on a leader is saved locally on the leader. • **write memory local** on a normal node is saved locally on the node.
terminal	Displays the commands necessary to create the running configuration. For example: **(config) # write terminal**

The following table summarizes other commands related to the **write** command:

Task	Command
Displays system memory utilization.	**# show memory**

GigaVUE-OS CLI—Configuration Examples

This chapter provides examples of how to configure the different features using the GigaVUE-OS CLI. Refer to the following sections:

- Configure Flow Mapping®
- Configure Active Visibility
- Configure GigaStream
- Configure Ingress and Egress VLAN
- Configure Inline Bypass Solutions
- Configure Inline Bypass Solution on GigaVUE-OS TAP Modules
- Configure Flexible Inline Arrangements
- Configure Inline SSL Decryption
- Configure GigaSMART Operations
- Configure Clustering
- Configure Multi-Path Leaf and Spine
- Configure GigaVUE HC Series Security Options

Configure Flow Mapping®

Flow Mapping® is the technology found in GigaVUE-OS nodes that takes line-rate traffic at 1Gb, 10Gb, 40Gb, or 100Gb from a network TAP or a SPAN/mirror port (physical or virtual) and sends it through a set of user-defined map rules to the tools and applications that secure, monitor, and analyze IT infrastructure.

The configuration examples for Flow Mapping is described in the following sections:

- How to Create Maps
- Configure Shared Collector Maps
- Map Priority
- Adjust Map Priority
- Packets Matching Multiple Rules in Same Map Example
- How to Add Comments to Map Rules
- Mixing Pass and Drop Rules
- Port Aliases
- User-Defined Pattern Match Examples
- How to Handle Overlaps when Sending VLANs and Subnets to Different Tools

- How to Create Map Rules for RTP Traffic
- IPv4 Criteria with GigaSMART Operation
- MAC Address Criteria with GigaStream
- IPv6 Criteria
- How to Create Map Rules with Inner Header Qualifiers and MPLS Header Qualifiers

Related Topics

- Refer to the "*Managing Maps*" section in the *GigaVUE Fabric Management Guide* for information about creating and managing a map.
- Refer to the map in the reference section for details on the syntax of the flow map CLI command.

How to Create Maps

You can create maps in the CLI using the prefix mode. The prefix mode lets you add map components on a new line, adding ingress ports, egress ports, GigaSMART operations, pass rules, and drop rules.

TIP: At any point in the prefix mode, you can use the **?** command to see the list of available commands and criteria you can use to define your map. Use of the ? will help you build maps easily without syntax errors.

You enter the map prefix mode with the following command:

```
(config) # map alias <alias>
```

After entering the map prefix mode, the system prompt changes to include the map alias you just specified. So, for example, if you entered the map prefix mode with **(config) # map alias vpntraffic**, the system prompt would update to include the **vpntraffic** alias as follows:

```
(config map alias vpntraffic) #
```

You can use this technique either to create an entirely new map or edit an existing map's settings, depending on whether the alias provided matches an existing map. Using the **?** technique will help you see the existing maps available for editing, as illustrated in Figure 1Using the ? Mark Technique with the Map Prefix Mode.

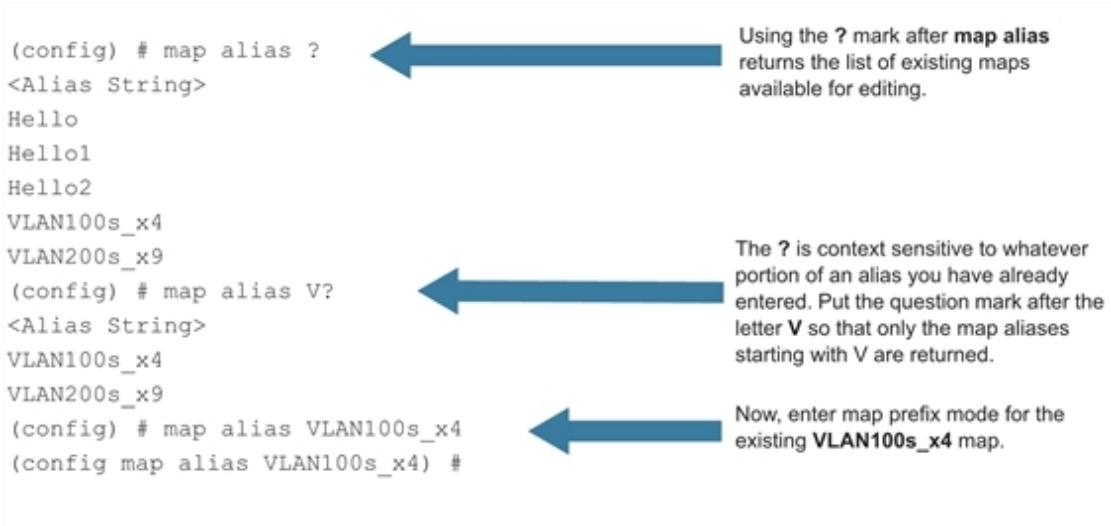


Figure 1 *Using the ? Mark Technique with the Map Prefix Mode*

You can use the **exit** command to exit the map prefix mode. The changes only take effect when you exit.

NOTE: It is recommended that you do not rename the maps with a prefix, "FM-Auto" through GigaVUE-OS CLI to avoid issues in using the Fabric maps solution.

Configure Shared Collector Maps

A shared collector is a special type of map configured with only a set of **from** ports and **shared-collector** ports or GigaStream. Rules, priority settings, GigaSMART operations and **to** destinations are not allowed in shared-collector maps

The following example shows the standard components of a shared-collector map. The from the ports match those used by a set of normal flow maps. The collector ports are where you want to send any packets not matching the normal flow maps.

Notice that the map does not include any rules, priority settings, GigaSMART operations (use gsop lines), or to destinations - those are not allowed in shared-collector maps.

Description	Command
First, enter the map prefix mode by specifying the alias for the shared-collector map.	**(config) # map-scollector alias shared_ collector**
We want to set the shared-collector map on the same network ports shared by the maps shown in Figure 6Existing Map on Network Port 1/2/x1.	**(config map-scollector alias shared_ collector) # from 14/4/x10**
Next, specify the shared-collector destination. You can	**(config map-scollector alias shared_**

Description	Command
configure the destination ports using standard port-list conventions. **Note**: **You could also specify a GigaStream as the shared-collector destination**.	**collector) # collector 14/6/x8**
Exit the map prefix mode.	**(config map-scollector alias shared_ collector) # exit**

Once the shared collector is set up, the CLI will display the newly configured map using the **show map** command.

To set a GigaStream as the shared collector, once the network port is set, use:

(config map-scollector alias shared_gscollector) # collector GigaStreamAlias1

To set multiple GigaStream as the shared collector, once the network port is set, use:

(config map-scollector alias shared_gscollector) # collector GigaStreamAlias1, GigaStreamAlias2, GigaStreamAlias3

Multiple tool ports can also be set as shared collector using the tool ID or range.

Configure Map-Passall

Description	Command
First, enter the map prefix mode by specifying the alias for the map-passall.	**(config) # map-passall alias MapPassAll**
We want to set the map-passall map on the same network ports shared by the maps shown in Figure 6Existing Map on Network Port 1/2/x1.	**(config map-passall alias MapPassAll) # from 14/4/x20**
Next, specify the map-passall destination. You can configure the destination ports using standard port-list conventions.	**(config map-passall alias MapPassAll) # to 14/6/x12**
Exit the map prefix mode.	**(config map-passall alias MapPassAll) # exit**

NOTE: Once the map-passall is set up, the CLI will display the newly configured map using the **show map** command.

Related Topics

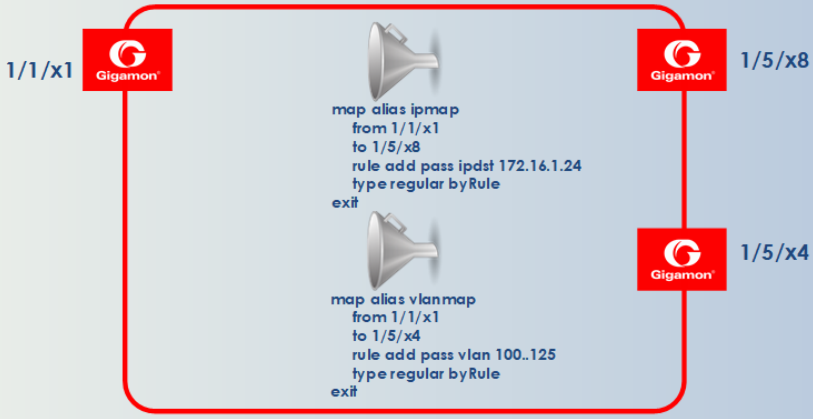
- Refer to *map-passall on page 1276* the reference section for details on the syntax of the flow map CLI command.
- Refer to *map-scollector on page 1276* the reference section for details on the syntax of the flow map CLI command.

Map Priority

Packets matching multiple maps in a configuration are sent to the map with the highest priority when the network ports are shared among multiple maps with pass-by map rules. By default, the first map configured has the highest priority; however, you can adjust this. The following figure graphically represents map priorities in Flow Mapping®:

How Classic Mode Handles Packets Matching Multiple Maps

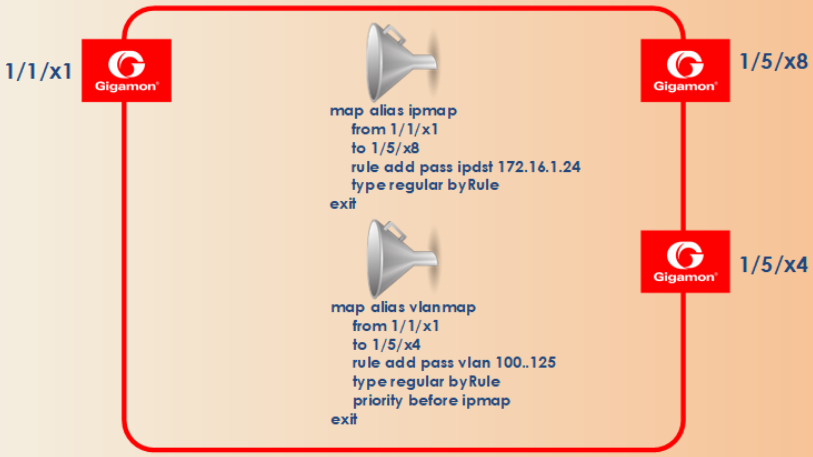
When operating in Classic Mode, a packet matching multiple maps is sent to the destination(s) specified by the map with the highest priority.
In this example, two maps are sending different data from Network Port 1/1/x1 to two different Tool Ports. The maps are listed in the same order in which they were created – **ipmap** first and **vlanmap** second. Because **vlanmap** was configured last, it has the lowest priority when operating in Classic Mode. In this case, a packet with a VLAN ID of 100 and a destination IP address of 172.16.1.24 would **only** be sent to 1/5/x8 because **ipmap** has the highest priority.



Adjusting Priority in Classic Mode

In some cases, you may not want to accept the default map priority assigned because of the order in which maps were created. In cases like this, you can *adjust map priority*.

For example, if we wanted to make sure that packets matching both **vlanmap** and **ipmap** were sent to the tool ports specified by **vlanmap**, we could explicitly assign priorities that made this happen. Notice that we've added a new line to **vlanmap** that ensures it has a higher priority than **ipmap** – **priority before ipmap.** With this line in place, packets with a VLAN ID of 100 and a destination IP address of 172.16.1.24 will be sent to 1/5/x4 because **vlanmap** has a higher priority than **ipmap**.



Maps sharing the same source port list are grouped together for the purpose of prioritizing their rules. Traffic is subjected to the rules of the highest priority map first and then the rules of the next highest priority map and so on. Within a map, drop rules are applied first and then pass rules, in other words, drop rules always have higher priority than pass rules.

Currently when a map's source port list is defined the map is grouped/prioritized with other maps sharing the same source port list. Newly configured maps are added as the lowest priority map within the group when initially configured unless changed by the user.

- The command **show map all** displays maps within a group top to bottom from highest priority to lowest priority.
- The command **show map priority** displays map order lists from left to right, highest priority to lowest priority
- The command **show map priority alias <map_alias>** displays the associated map order list (if any) from left to right, highest priority to lowest priority.

NOTE: The command **show running-config** displays the map rule IDs but not in the same order in which it was configured.

NOTE: A shared collector will always go to the lowest priority when setting up maps.

Adjust Map Priority

Before you get started adjusting map priority, start by reviewing the current map priorities in place with the **show map priority** command.

Then, once you have reviewed the existing hierarchy of map priorities, you can fine-tune the priority of maps by using the **map alias <alias> priority** command with the following arguments:

map alias <alias> priority after Set map priority after another map. before Set map priority before another map. highest Set map to highest priority lowest Set map to lowest priority

For example, the following command sets the map with the alias of **bigmap** to the highest priority:

```
(config) # map alias bigmap priority highest
```

The **priority** argument is also available in the prefix mode. For example, the following commands enter the prefix mode for the existing map with the alias of **bigmap** and then set its priority after a second map called IP25:

```
(config) # map alias bigmap
(config map alias bigmap) # priority after IP25
(config map alias bigmap) # exit
(config) #
```

Packets Matching Multiple Rules in Same Map Example

Figure 2Packet Matching Multiple Rules in Same Map illustrates how Flow Mapping® handles a case where a packet matches multiple rules in the same map. In cases like this, the packet is sent to all configured destinations when the first pass rule is matched (assuming there were no matching drop rules – drop rules have higher priority). You can see the port statistics illustrating this in Figure 3Show Port Stats Output for Figure 2Packet Matching Multiple Rules in Same Map.

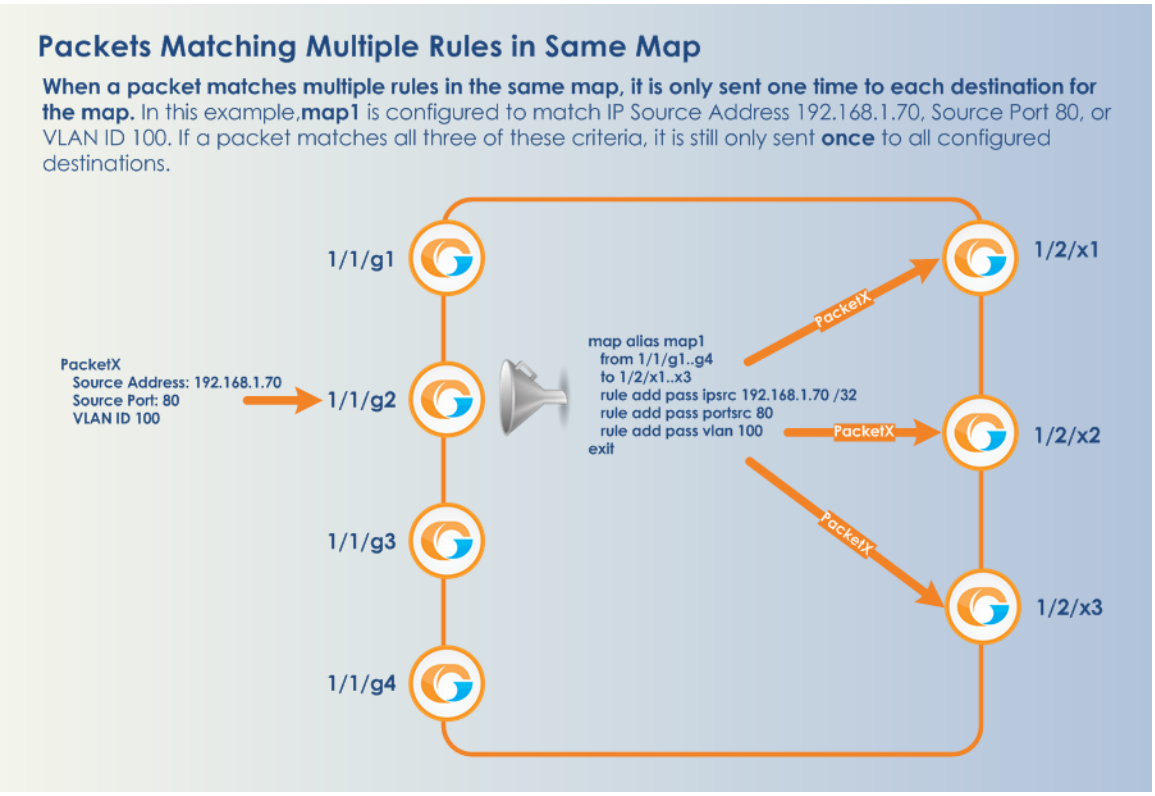


Figure 2 *Packet Matching Multiple Rules in Same Map*

```
(config) # show port stats port-list 1/2/x1..x3,1/1/g1..g4

       Counter Name      Port: 1/2/x1      Port: 1/2/x2      Port: 1/2/x3      Port: 1/1/g2
 ================== ================= ================= ================= =================
         IfInOctets:                 0                 0                 0             10000
      IfInUcastPkts:                 0                 0                 0                 0
     IfInNUcastPkts:                 0                 0                 0               100
       IfInPktDrops:                 0                 0                 0                 0
       IfInDiscards:                 0                 0                 0                 0
         IfInErrors:                 0                 0                 0                 0
   IfInOctetsPerSec:                 0                 0                 0                 0
 IfInPacketsPerSec:                  0                 0                 0                 0
        IfOutOctets:             10000             10000             10000                 0
     IfOutUcastPkts:                 0                 0                 0                 0
    IfOutNUcastPkts:               100               100               100                 0
      IfOutDiscards:                 0                 0                 0                 0
        IfOutErrors:                 0                 0                 0                 0
  IfOutOctetsPerSec:                 0                 0                 0                 0
IfOutPacketsPerSec:                  0                 0                 0                 0
```

Figure 3 *Show Port Stats Output for Figure 2Packet Matching Multiple Rules in Same Map*

Port Lists

Many map commands require a port-list (for example, rule and shared-collector arguments all require them). You can define the port lists using any combination of the standard conventions:

port-id <bid/sid/pid>

port-alias <port-alias>

port-list <bid/sid/pid_x..pid_y> (range) | <bid/sid/pid_x,bid/sid/pid_y,bid/sid/pid_z> (list)

- In the **from** argument of the **map-passall** command, you can specify a network port list or an inline network alias. In the **to** argument of the **map-passall** command, you can specify a tool port list, an inline tool alias, an inline tool group alias, or an inline bypass.
- The **port-list** argument lets you select multiple non-contiguous ports. To enter port IDs in a list, simply put a comma between each port ID in the list.
- The **<bid/sid/pid_x..pid_y>** argument lets you select a series of adjacent ports (for example, **1/5/x4..x6** selects port x4..x6 on slot 5.

  NOTE: Port ranges must be specified separately for 10Gb-capable and 1Gb ports – you cannot create a single range including both. For example, the PRT-H00-X12G04 card includes ports x1..x12 and ports g1..g4, but you cannot create a series that spans from **1/1/x1** to **1/1/g4**. Instead, you must create two series – **1/1/x1..x12** and **1/1/g1..g4**.

- GigaSMART load balancing port groups can have ports with different rates.
- You can mix a **port-id** with a **port-alias** and a **port-list** so long as they are separated by commas and no spaces. For example, **1/5/x4..x6,myalias,1/4/x2..x4** is a valid port-list.

Configure Port Aliases

The GigaVUE-OS lets you configure textual aliases for network and tool ports. Aliases can be used in place of the numerical **bid/sid/pid** identifier required in many packet distribution commands in the CLI. For example, instead of configuring a map from 1/1/x1 to 1/1/x4, you could map **Gb_In** to **Stream-to-Disk**; the following table shows the commands to configure these aliases and the map itself.

Command	Comments
(config) # port 1/1/x1 alias Gb_In	Configures port 1/1/x1 with the alias of **Gb_In**.
(config) # port 1/1/x4 type tool	Sets port x4 on slot 1 (1/1/x4) as a tool port.
(config) # port 1/1/x4 alias Stream-to-Disk	Configures port 1/1/x4 with the alias of Stream-to-Disk
(config) # map alias GbCnx **(config map alias GbCnx) # from Gb_In**	Creates a regular map passing all IPv4 traffic between 1/1/x1 and 1/1/x4 using their aliases of Gb_In and

Command	Comments
(config map alias GbCnx) # to Stream-to-Disk (config map alias GbCnx) # rule add pass ipver 4 **(config map alias GbCnx) # type regular byRule** **(config map alias GbCnx) # exit**	Stream-to-Disk, respectively.

How to Add Comments to Map Rules

You can add comments to map rules. Use comments to label the purpose of a rule or the type of traffic covered by a rule.

You can add comments to the following CLI **map** commands:

- map rule add pass
- map rule add drop
- map gsrule add pass
- map gsrule add drop
- map flowrule add pass gtp
- map flowrule add drop gtp

Map rule comments are optional on these commands.

Consider the following when adding map rule comments:

- Use up to 128 characters, including spaces.
- Enclose the comment in quotation marks, if the comment is longer than one word.
- To include double quotation marks (") inside the quotation marks, precede it with a backslash (\).

Map Rule Comments Examples

The following are individual examples of valid map rule comments.

Command
(config) # map alias 1 rule add pass ipver 4 comment “Allow IPv4”
(config) # map alias 1 rule add pass macdst 11:22:33:44:55:66 00:00:00:00:00:00 comment “Allow MAC dst 11:22:33:44:55:66”
(config) # map alias 1 rule add drop ipver 6 comment “Drop IPv6”
(config) # map alias 1 rule add drop macdst 11:22:33:44:55:66 00:00:00:00:00:00 comment “Drop MAC dst 11:22:33:44:55:66”
(config) # map alias 1 gsrule add pass ipver any value 4 comment "Allow IPv4"

Command
(config) # map alias 1 gsrule add drop ipv6 flow-label any value 6 comment "Drop IPv6"
(config) # map alias 1 flowrule add pass gtp imsi 123456 comment "Allow imsi 123456"
(config) # map alias 1 flowrule add drop gtp imsi 123456 comment "Drop imsi 123456"
(config) # map **alias 1** rule add pass ipver 4 comment “Allow \“IPv4\””

NOTE: Error messages are displayed when a comment is invalid, for example:

- if the comment is longer than one word and does not include double quotation marks
- if the comment is longer than 128 characters
- if the rule with which the comment is included is not valid

Map Rule Logic Examples

For example, the rules shown in the following table are both set up with criteria for **vlan 100** and **portsrc 23**.

- The first example combines the two criteria into a single rule. This joins the criteria with a logical **AND**.
- The second example creates two separate rules – one for each of the criteria. This joins the criteria with a logical **OR**.

	CLI Commands	Description
Criteria in Single Rule Joined with AND	**(config map alias mymap) # rule add pass vlan 100 portsrc 23**	Creates single rule with two criteria – VLAN ID **100** and source port **23**. Because the criteria are in a single rule, they are joined with a logical AND. This means that a packet must match both **VLAN 100 and portsrc 23** to match this rule.
Multiple Rules Joined with OR	**(config map alias mymap) # rule add pass vlan 100**	Creates a rule in the map called **mymap** for VLAN ID 100.
	(config map alias mymap) # rule add pass portsrc 23	Creates a rule in the map called **mymap for** source port **23**. Because the criteria are in separate rules, they are joined with a logical OR. This means that a packet can match either **VLAN 100 or portsrc 23** to match this map.

Mixing Pass and Drop Rules

GigaVUE-OS lets you mix pass and drop rules on a single port. Mixing pass and drop rules can be useful in a variety of situations. The following example shows a pass rule set up to include all traffic matching a particular source port range combined with a drop rule configured to exclude ICMP traffic.

Description	CLI Commands
Enters the map prefix mode for a map named mymap.	(config) map alias mymap
Specifies the map type and subtype	(config map alias mymap) # type regular byRule
Specifies that this map will apply to traffic arriving on network port 1/1/x5.	(config map alias mymap) # from 1/1/x5
Specifies that packets matching this map will be sent to tool port 3/5/x5.	(config map alias mymap) # to 3/5/x5
Create a rule that will match all packets with a source port between 20..66.	**(config map alias mymap) # rule add pass portsrc 20..66**
Create a rule that will drop all ICMP-IPv4 traffic.	**(config map alias mymap) # rule add drop protocol icmp-ipv4**
Exits the map prefix mode.	(config map alias mymap) # exit

Drop Rules Have Precedence!

Keep in mind that within a map, drop rules have precedence over pass rules. So, if a packet matches both a pass and a drop rule in the same map, the packet is dropped rather than passed.

Port Aliases

GigaVUE-OS lets you configure textual aliases for network and tool ports. Aliases can be used in place of the numerical **bid/sid/pid** identifier required in many packet distribution commands in the CLI. For example, instead of configuring a map between, say, 1/1/x1 and 1/1/x4, you could map from **Gb_In** to **Stream-to-Disk**; the following table shows the commands to configure these aliases and the map itself.

Command	Comments
(config) # port 1/1/x1 alias Gb_In	Configures port 1/1/x1 with the alias of **Gb_In**.
(config) # port 1/1/x4 type tool	Sets port x4 on slot 1 (1/1/x4) as a tool port.

Command	Comments
(config) # port 1/1/x4 alias Stream-to-Disk	Configures port 1/1/x4 with the alias of Stream-to-Disk
(config) # map alias Gbmap **(config map alias Gbmap) # type regular byRule** **(config map alias Gbmap) # from Gb_In** **(config map alias Gbmap) # to Stream-to-Disk** **(config map alias Gbmap) # exit**	Creates a regular map from 1/1/x1 to 1/1/x4 using their aliases of Gb_In and Stream-to-Disk, respectively. Notice that this map does not have any rules – the full stream of packets arriving on 1/1/x1 will be sent to 1/1/x4.

User-Defined Pattern Match Rules

GigaVUE-OS lets you create pass and drop map rules with *pattern matches* to search for a particular sequence of bits at a specific offset in a packet. You can configure up to two user-defined, 16-byte **pattern matches** in a map rule. A **pattern** is a particular sequence of bits at a specific location in a frame.

NOTE: Refer to User-Defined Pattern Match Examples for step-by-step instructions on creating a real-world pattern-match map rule.

NOTE: The CLI refers to a pattern as a **UDA** ("user-defined attribute").

User-defined pattern matches consist of the following components:

Step	Description
Pattern	Use the **uda1-data** and **uda2-data** arguments for map rule commands to set up the actual bit patterns you want to search for. Refer to User-Defined Pattern Match Examples for details.
Mask	Use the **uda1-mask** and **uda2-mask** arguments for map rule commands to specify which bits in the pattern must match to satisfy the map rule.
Offset	Use the **uda1-offset** and **uda2-offset** arguments for map rule commands to specify where in the packet bits must match. **NOTE:** A maximum of two offsets per GigaVUE HC Series line card, module, or node orGigaVUE TA Series node are accepted. When both of the available offsets for a line card, module, or node are in use with existing map rules, you will not be able to add a new rule with a different value for **udax-offset** until at least one of the **udax-offsets** is freed up from all existing map rules.

User-Defined Pattern Match Syntax

The user-defined pattern match syntax is as follows:

[uda1-data <16-byte-hex>] [uda1-mask1 <16-byte-hex>][uda1-offset <2~110 bytes>][uda2-data <16-byte-hex>] [uda2-mask2 <16-byte-hex>][uda2-offset <2~110 bytes>]

- Both the **uda*x*-data** and **uda*x*-mask** arguments are specified as 16-byte hexadecimal sequences. Specify the pattern in four 4-byte segments separated by hyphens. For example:

  0x01234567-89abcdef-01234567-89abcdef
- Masks specify which bits in the pattern must match. The mask lets you set certain bits in the pattern as wild cards – any values in the masked bit positions will be accepted.
  - Bits masked with binary 1s must match the specified pattern.
  - Bits masked with binary 0s are ignored.
- You can set up the two global offsets allowed per GigaVUE HC Series line card, module, or node or GigaVUE TA Series node at 4-byte boundaries beginning at frame offset 2 and ending at offset 110. The resulting data range for pattern matches is from byte 3 through byte 126.
  - Multiple offsets must be set either equal to one another, or set beyond the boundaries of each other. For example, if **uda1-offset** starts at byte 2, the **uda2-offset** can only start either at byte 2 or at any point beginning with byte 18 (which would be the next 4-byte boundary after the 16-byte pattern used at **uda1-offset**).
  - Offsets are always frame-relative, not data-relative.
  - In many cases, you will be looking for patterns that do not start exactly on a four-byte boundary. To search in these position, you would set an offset at the nearest four-byte boundary and adjust the pattern and mask accordingly.
- On GigaVUE-TA400, and GigaVUE-TA400E both UDA1, UDA2 support 8-byte data match from offset 0 to 160 on a 4-byte boundary. UDA1 offset value specifies offset from start of ethertype, UDA2 offset value specifies offset from start of L3 header. UDA1, UDA2 offset overlap is allowed. UDA rule having both GigaVUE-TA400 and non GigaVUE-TA400 source ports in a map is not allowed. The data to be matched must be specified with trailing 8-byte zeros. For example to specify 0x54206874 as a match value, enter 00000000-54206874-00000000-00000000.

User-Defined Pattern Match Rules

Keep in mind the following rules when creating user-defined pattern matches:

- Offsets are specified in decimal; patterns and masks are specified in hexadecimal.
- All hexadecimal values must be fully defined, including leading zeroes. For example, to specify 0xff as a 16-byte value, you must enter 00000000-00000000-00000000-000000ff.
- User-defined pattern-match criteria are not allowed in egress port-filters.

- You can use user-defined pattern matches as either standalone map rules or in tandem with the other available predefined criteria for map rules (for example, port numbers, IP addresses, VLAN IDs, and so on).
- You can use up to two separate user-defined pattern matches in a single map rule. When two user-defined pattern matches appear in the same map rule, they are joined with a logical AND. However, the two patterns cannot use the same offset.
- User-defined pattern matches are combined in map rules using the same logic described in the *"Combining Rules and Rule Logic"* section in the *GigaVUE Fabric Management Guide*.
- Avoid using user-defined pattern matches to set map rules for elements that are available as predefined criteria (for example, IP addresses, MAC addresses, and so on).
- A maximum of two offsets per GigaVUE HC Series line card, module, or node or GigaVUE TA Series node are accepted. When both of the available offsets for a line card, module, or node are in use with existing map rules, you will not be able to add a new rule with a different value for **udax-offset** until at least one of the **udax-offsets** is freed up from all existing map rules.
- You cannot combine user-defined patterns with certain qualifiers in the same rule. This is due to hardware limitations.
- On GigaVUE-HC1-Plus, GigaVUE-HCT, GigaVUE-TA25, and GigaVUE-TA25E, UDA1 supports a 12-byte data match of tagged traffic with an offset starting from 16 to 116. UDA1 does not support untagged traffic. UDA2 supports a 4-byte data match with an offset starting from 0 to 60.
- On GigaVUE-TA400, both UDA1, UDA2 support 8-byte data match from offset 0 to 160 on a 4-byte boundary. UDA1 offset value specifies offset from start of ethertype, UDA2 offset value specifies offset from start of L3 header. UDA1, UDA2 offset overlap is allowed. UDA rule having both GigaVUE-TA400 and non GigaVUE-TA400 source ports in a map is not allowed. The data to be matched must be specified with trailing 8-byte zeros. For example to specify 0x54206874 as a match value, enter 00000000-54206874-00000000-00000000.
- Due to limitations in the platform, map rules to match the data pattern on Inner-L3 and Inner-L4 qualifiers with UDA base does not match the Inner headers for the following types of encapsulation:
    - ERSPAN
    - LISP
    - L2GRE

  However, the inner headers of ERSPAN, LISP, L2GRE can be made to match with outer L3 and Outer L4 UDA base by appropriately adjusting the offsets.
    - When configuring a map rule with common UDA offset, you cannot combine source ports from GigaVUE-HC1-Plus or GigaVUE-TA25 devices together with source ports from other devices in a single map. You must create separate maps for GigaVUE-HC1-Plus or GigaVUE-TA25 devices and other devices.

User-Defined Pattern Match Examples

In this example, a 3G carrier is monitoring the Gn interface between the SGSN and the GGSN in the mobile core network and wants to split traffic from different subscriber IP address ranges to different tool ports. However, because the subscriber IP addresses are tunneled using the GPRS Tunneling Protocol (GTP), standard IP address map rules will not work. The addresses are always at the same offsets, though, so we can construct UDA pattern match rules to match and distribute the traffic correctly.

For example, suppose we want to apply the following rules to all traffic seen on network port 1/5/x1:

- Send all traffic to and from the 10.218.0.0 IP address range inside the GTP tunnel to tool port 1/5/x4.
- Send all traffic to and from the 10.228.0.0 IP address range inside the GTP tunnel to tool port 1/5/x9.

Keep in mind that we also know the following about tunneled GTP traffic:

- The offset for source IP addresses inside the GTP tunnel is 62.
- The offset for destination IP addresses inside the GTP tunnel is 66.

The following example explains how to construct two maps that will distribute traffic using UDA pattern match rules.

Description	Command
Map #1 – GTP_Map218 Our first map will send traffic to and from the 10.218.0.0 IP address range inside the GTP tunnel to tool port 1/5/x4.	
Start by entering the prefix mode for a new map called **GTP_Map218**.	(config) # map alias GTP_Map218
Specifies the map type and subtype.	(config map alias GTP_Map218) # type regular byRule
Specify that this map will match packets arriving on network port 1/5/x1.	(config map alias GTP_Map218) # from 1/5/x1
Specify that packets matching this map will be sent to tool port 1/5/x4.	(config map alias GTP_Map218) # to 1/5/x4
Next, add the map rules for our first address range – 10.218.0.0. This IP address translates to 0ada in hex. The first rule matches the 10.218.0.0 address at the source address offset of 62 in the GTP tunnel.	(config map alias GTP_Map218) # rule add passuda1-data 0ada0000-00000000-00000000-00000000 uda1-mask ffff0000-00000000-00000000-00000000 uda1-offset 62

Description	Command
The second rule matches the same address range (10.218.0.0) but at the destination address offset of 66 in the GTP tunnel. Notice that we have still specified the offset as 62 and have simply masked out to the correct location of the destination address. This way, we have still only used one of the two possible offsets in place for the GigaVUE HC Series node at any one time.	(config map alias GTP_Map218) # rule add passuda1-data 00000000-0ada0000-00000000-00000000 uda1-mask 00000000-ffff0000-00000000-00000000 uda1-offset 62
Exit the map prefix mode.	(config map alias GTP_Map218) # exit
Map #2 – GTP_Map228 Our second map will send traffic to and from the 10.228.0.0 IP address range inside the GTP tunnel to tool port 1/5/x9.	
Start by entering the prefix mode for a new map called **GTP_Map228**.	(config) # map alias GTP_Map228
Specifies the map type and subtype.	(config map alias GTP_Map228) # type regular byRule
Specify that this map will match packets arriving on network port 1/5/x1.	(config map alias GTP_Map228) # from 1/5/x1
Specify that packets matching this map will be sent to tool port 1/5/x9.	(config map alias GTP_Map228) # to 1/5/x9
Now, create rules for the second address range – 10.228.0.0 (0ae4 in hex). As with the first range, create separate rules for the source and destination offsets inside the GTP tunnel. This address range is being sent to 1/1/x4.	(config map alias GTP_Map228) # rule add passuda1-data 0ae40000-00000000-00000000-00000000 uda1-mask ffff0000-00000000-00000000-00000000 uda1-offset 62
Here is the companion rule for the destination address offset of 66.	(config map alias GTP_Map228) # rule add passuda1-data 00000000-0ae40000-00000000-00000000 uda1-mask 00000000-ffff0000-00000000-00000000 uda1-offset 62
Exit the map prefix mode.	(config map alias GTP_Map228) # exit

Map Examples

This section provides the following map examples:

- How to Handle Overlaps when Sending VLANs and Subnets to Different Tools
- How to Create Map Rules for RTP Traffic
- How to Use MAC Address/Mask Map Rules
- IPv4 Criteria with GigaSMART Operation
- MAC Address Criteria with GigaStream
- IPv6 Criteria

- UDA Pattern Match Criteria
- Null Port in Maps

In addition, refer to the following sections for more examples of creating maps:

- User-Defined Pattern Match Examples

How to Handle Overlaps when Sending VLANs and Subnets to Different Tools

Figure 4Sending Subnets and VLANs to Different Ports shows how to use map priority when handling packets matching criteria in multiple maps. In this example, we want to achieve the following results:

- Send packets on the 172.16.0.0 subnet to 1/2/x1
- Send packets on the 172.17.0.0 subnet to 1/2/x2
- Send packets on VLAN 100 to 1/2/x3

The trick is in how to handle packets on either 172.16.0.0 or 172.17.0.0 **and** VLAN 100. In this example, we use map priority to ensure that packets such as this are sent to both of their desired destinations.

Notice that the first two maps configured in Figure 4Sending Subnets and VLANs to Different Ports.

The same principle is applied in **map2** for packets on 172.17.0.0 and VLAN 100.

NOTE: If we did not observe the order of map entry shown in Figure 4Sending Subnets and VLANs to Different Ports.

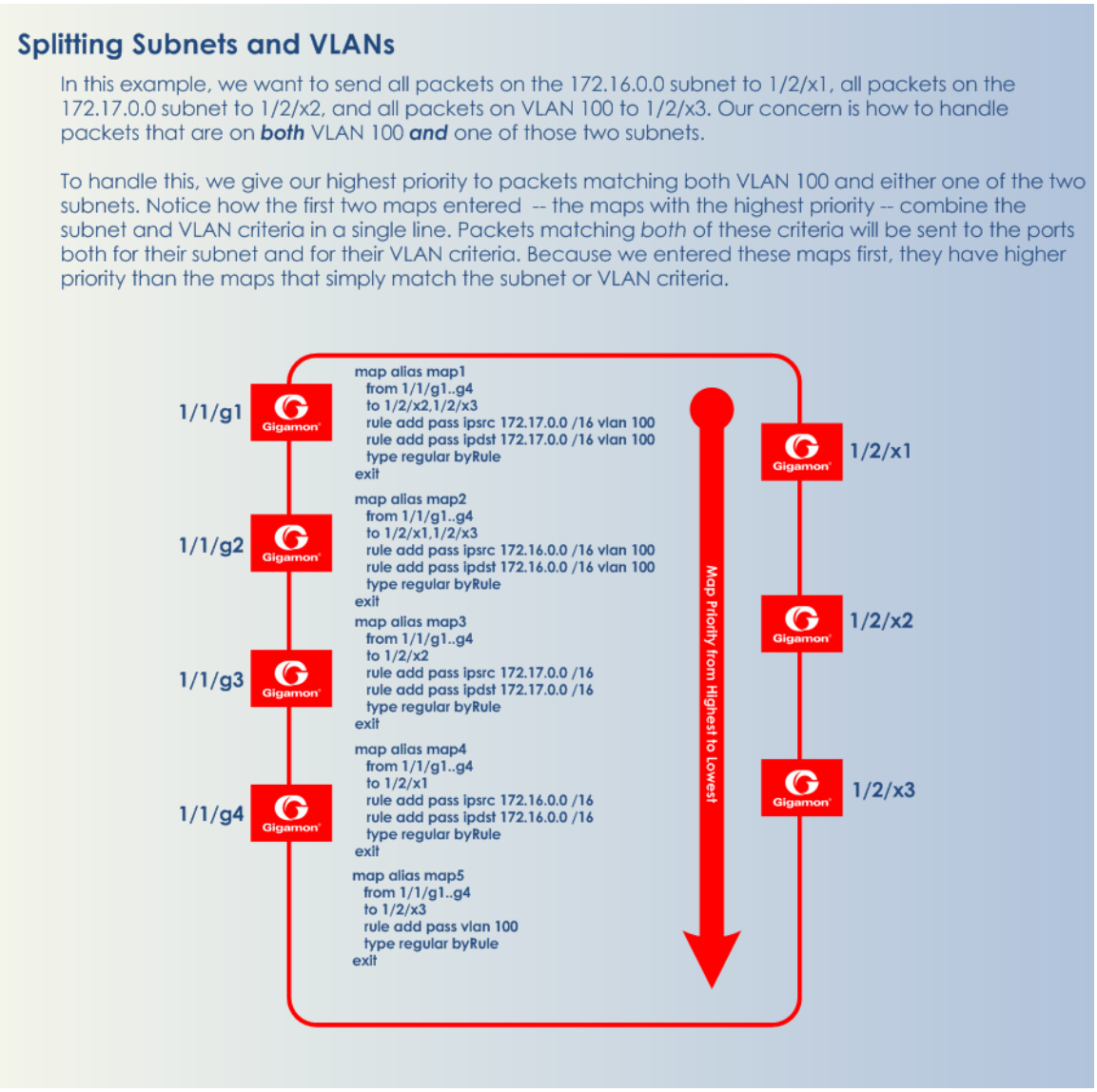


Figure 4 *Sending Subnets and VLANs to Different Ports*

How to Create Map Rules for RTP Traffic

You can use GigaVUE-OS to set map rules matching even or odd port numbers to focus on different aspects of VoIP traffic.

VoIP implementations typically send RTP on even port numbers and RTCP on the next available odd port number. The following example constructs a map on network ports 1/4/x7 and 1/4/x2 with map rules designed to block RTP on the even-numbered ports in its common ranges.

Table 1: Blocking RTP Traffic on Common Ports

Command	Description
(config) # map alias no_rtp	Enters the map prefix mode for a new map with the alias of no_rtp.
(config map alias no_rtp) # type regular byRule	Specifies the map type and subtype.
(config map alias no_rtp) # from 1/4/x1..x2	Applies the map to traffic arriving on network ports 1/4/x1 and 1/4/x2.
to 1/5/x12	Sends matching traffic to tool port 1/5/x12
rule add drop portsrc 5004	Constructs a rule to drop traffic with a source port of 5004.
rule add drop portdst 5004	Constructs a rule to drop traffic with a destination port of 5004.
rule add drop portsrc 16384..16624 portsrc-subset even	Constructs a rule to drop traffic with an even-numbered source port in the range of 16384..16624. This is a standard RTP port range used by Cisco equipment.
rule add drop portdst 16384..16624 portdst-subset even	Constructs a rule to drop traffic with an even-numbered source port in the range of 16384..16624.
shared-collector 1/5/x8	Sends non-matching traffic to a shared-collector on 1/5/x8. This shared-collector will be used for packets not matching any maps on network ports 1/4/x1..x2; refer to the *"About Shared Collectors"* section in the *GigaVUE Fabric Management Guide* for details.
(config map alias no_rtp) # exit	Exits the map prefix mode.
(config) # write memory	Saves changes to the active configuration file.

Use the **show map brief** and **show map** command output to check the progress.

How to Use MAC Address/Mask Map Rules

This section provides several examples of how to use MAC address rules with an address mask.

Example 1 – Drop Rule

In this example, set up a map rule that denies packets with a source MAC address matching that specified in the map rule. The map rule will use the following values for **macsrc** and **<mac-netmask>**:

Field in Map Rule Command	Value
macsrc	00 00 00 00 00 03
<mac-netmask>	FF FF FF FF FF FE

Command:

(config map macmap) # rule add drop macsrc 00:00:00:00:00:03 ffff.ffff.fffe

Result:

Packets with the following two MAC source addresses are dropped:

- 00:00:00:00:00:02
- 00:00:00:00:00:03

All other MAC addresses will pass this filter.

Example 2 – Pass Rule

In this example, we will change the map rule action we set up in Example 1 – Drop Rule from **drop** to **pass**.

Command:

(config map passmac) # rule add pass macsrc 00:00:00:00:00:03 ffff.ffff.fffe

Result:

Only packets with the following two MAC source addresses are accepted:

- 00:00:00:00:00:02
- 00:00:00:00:00:03

All other MAC addresses are denied.

Example 3 – Drop Rule

In this example, set up a map rule that denies packets with a source MAC address matching that specified in the map rule. The map rule will use the following values for **macsrc** and **<mac-netmask>**:

Field	Value
macsrc	00:00:00:00:00:03
<mac-netmask>	FFFF.FFFF. FFF1

Command:

(config map macdrop) # rule add drop macsrc 00:00:00:00:00:03 ffff.ffff.fff1

Result:

Packets with the following eight MAC source addresses are dropped:

- 00:00:00:00:00:01

- 00:00:00:00:00:03
- 00:00:00:00:00:05
- 00:00:00:00:00:07
- 00:00:00:00:00:09
- 00:00:00:00:00:0b
- 00:00:00:00:00:0d
- 00:00:00:00:00:0f

All other MAC addresses will pass this map rule.

Example 4 – Dropping Odd-Numbered MAC Addresses

In this example, set up a rule that denies packets with a source MAC address matching that specified in the map rule. The map rule will use the following values for **macsrc** and **<mac-netmask>**:

Field	Value
macsrc	00:00:00:00:00:03
<mac-netmask>	0000.0000.0001

Command:

(config map oddmac) # rule add drop macsrc 00:00:00:00:00:03 0000.0000.0001

Result:

All odd-numbered MAC source addresses are denied:

- 00:00:00:00:00:01
- 00:00:00:00:00:03
- ff:ff:ff:ff:ff:fb
- ff:ff:ff:ff:ff:fd
- ff:ff:ff:ff:ff:ff

Only packets from even-numbered MAC source addresses will pass through this rule. All the odd-numbered MAC source addresses are dropped.

Example 5 – Allowing Odd-Numbered MAC Addresses

In this example, we will change the map rule action we set up in Example 4 – Dropping Odd-Numbered MAC Addresses from **drop** to **pass**.

Command:

(config map oddmac) # rule add pass macsrc 00:00:00:00:00:03 0000.0000.0001

Result:

Only packets from odd-numbered MAC source addresses will pass through this rule. All the even-numbered MAC source addresses are dropped.

Example 6 – Allowing All Traffic to Pass Through Based on Wild-card MAC Address

In this example, we will change the map rule action we set up a wild card MAC address for all traffic. This is useful when all traffic is required to go to the tool port but one cannot use the map-passall command because a GigaSMART operation is required on the traffic.

Command:

```
(config map passallmac) # rule add pass macsrc 00:00:00:00:00:00 00:00:00:00:00:00
```

Result:

All packets will pass through this rule to the tool port without filtering.

IPv4 Criteria with GigaSMART Operation

The following example creates a simple de-duplication GigaSMART operation and includes it in map with IPv4 source address and port criteria:

```
(config) # gsop alias dedup1 dedup set port-list gsgrp1
(config) # map alias gigamon1
(config map alias gigamon1) # type regular byRule
(config map alias gigamon1) # from 1/1/g1..g4
(config map alias gigamon1) # use gsop dedup1
(config map alias gigamon1) # to 1/2/x1..x2
(config map alias gigamon1) # rule add pass ipsrc 192.168.1.70 /32 portsrc 80
(config map alias gigamon1) # rule add pass ipdst 192.168.1.70 /32 portdst 80
(config map alias gigamon1) # rule add drop ethertype 0806
(config map alias gigamon1) # exit
(config) #
```

MAC Address Criteria with GigaStream

The following example creates a GigaStream and uses it as a destination for packets matching the specified MAC address and VLAN criteria:

```
(config) # gigastream alias gs1 port-list 1/2/x28..x32 params hash advanced
(config) # map alias gigamon2
(config map alias gigamon2) # type regular byRule
(config map alias gigamon2) # from 1/1/g1..g4
(config map alias gigamon2) # to 1/2/x4,1/2/x6,1/2/x8,gs1
(config map alias gigamon2) # rule add pass vlan 100 protocol tcp
(config map alias gigamon2) # rule add pass macsrc 00:11:22:33:44:55 ffff.ffff.ffff
(config map alias gigamon2) # exit
(config) #
```

IPv6 Criteria

The following example uses an IPv6 source address as a pass rule:

```
(config) # map alias gigamon3
(config map alias gigamon3) # type regular byRule
(config map alias gigamon3) # from 1/1/g1..g4
(config map alias gigamon3) # to 1/2/x4,1/2/x6,1/2/x8
(config map alias gigamon3) # rule add pass ip6src FE80:0:0:0:202:B3FF:FE1E:8329 /64
(config map alias gigamon3) # exit
(config) #
```

UDA Pattern Match Criteria

The following example uses UDA pattern match criteria as part of a pass rule:

```
(config) # map alias uda
(config map alias uda) # type regular byRule
(config map alias uda) # from 1/1/g1..g4
(config map alias uda) # to 1/2/x4,1/2/x6,1/2/x8
(config map alias uda) # rule add pass uda1-data 12345678-12345678-12345678-12345678 uda1-mask 0000ffff-0000ffff-0000ffff-0000ffff uda1-offset 10
(config map alias uda) # exit
(config) #
```

Null Port in Maps

The following example uses null-port in a regular map:

```
(config) # map alias map_null
(config map alias map-null) # type regular
(config map alias map-null) # from 1/2/x4
(config map alias <map-alias>) # to null-port
(config map alias <map-alias>) # use gsop gsop apf
(config map alias <map-alias>) # rule add pass/drop vlan 10
(config map alias <map-alias>) # exit
```

How to Create Map Rules with Inner Header Qualifiers and MPLS Header Qualifiers

To configure a Inner Header qualifier and MPLS Header qualifier in regular by-rule maps, refer to the following example:

NOTE: The custom filter template must be configured with Inner Header and MPLS Header attributes before it can be used in a map rule. Refer to the Filter Templates chapter in the *GigaVUE Fabric Management Guide*

Regular by-rule map

Map level configuration with Inner Header qualifier

Step	Description	Command
1.	configure flexible filter-template with inner and mpls qualifiers.	**(config) # filter-template alias ft1 qualifiers add inner-ipdst inner-ip6dst**
2.	Attach the flexible filter-template to the GigaVUE-TA400 pseudo-slot based on the network port.	**(config) # card slot 1/1PS1 filter-template ft1**
3.	Configure port 1/1/d1 with the type network.	**(config) # port 1/1/d1 type network**
4.	Configure port 1/1/d11 with the type tool.	**(config) # port 1/1/d11 type tool**
5.	Configure the map with the alias of m1.	**(config) # map alias m1**
6.	Configure the source port for map m1.	**(config map alias m1) # from 1/1/d1**
7.	Configure the destination port for map m1.	**(config map alias m1) # to 1/1/d11**
8.	Configures rule 1 which qualifies the	**(config map alias m1) # rule add pass inner-**

Step	Description	Command
	inner IPv6 destination header.	**ip6dst 2000:4::4d /128**
9.	Exit map configuration.	**(config map alias m1) # exit**

Map level configuration with MPLS header qualifier

Step	Description	Command
1.	configure flexible filter-template with mpls qualifers.	**(config) # filter-template alias ft2 qualifiers add mpls-label-id pos 1 mpls-label-ttl pos 1**
2.	Attach the flexible filter-template to the GigaVUE-TA400 pseudo-slot based on the network port.	**(config) # card slot 1/1PS2 filter-template ft2**
3.	Configure port 1/1/d10 with the type network.	**(config) # port 1/1/d10 type network**
4.	Configure port 1/1/d11 with the type tool.	**(config) # port 1/1/d11 type tool**
5.	Configure the map with the alias of m2.	**(config) # map alias m2**
6.	Configure the source port for map m2.	**(config map alias m2) # from 1/1/d10**
7.	Configure the destination port for map m2.	**(config map alias m2) # to 1/1/d11**
8.	Configures rule 1 which qualifies the two MPLS headers.	**(config map alias m2) # rule add pass mpls-label-id 102240 pos 1 mpls-label-ttl 250 pos 1**
9.	Exit map configuration.	**(config map alias m2) # exit**

Related Topics:

- To configure VLAN manipulation using GigaVUE-FM, refer to the Inner Header and MPLS Header Filtering section in Flow Map Syntax and Construction chapter in the *GigaVUE Fabric Management Guide*.

map-passalls and port mirrors

How to Send All Traffic to IDS: map-passall

Intrusion Detection Systems need to see all traffic to work effectively. However, you may want to use maps to send different portions of the same traffic source to different destinations. This is the perfect place to use a map-passall. Figure 5Sending All Traffic to IDS

illustrates this:

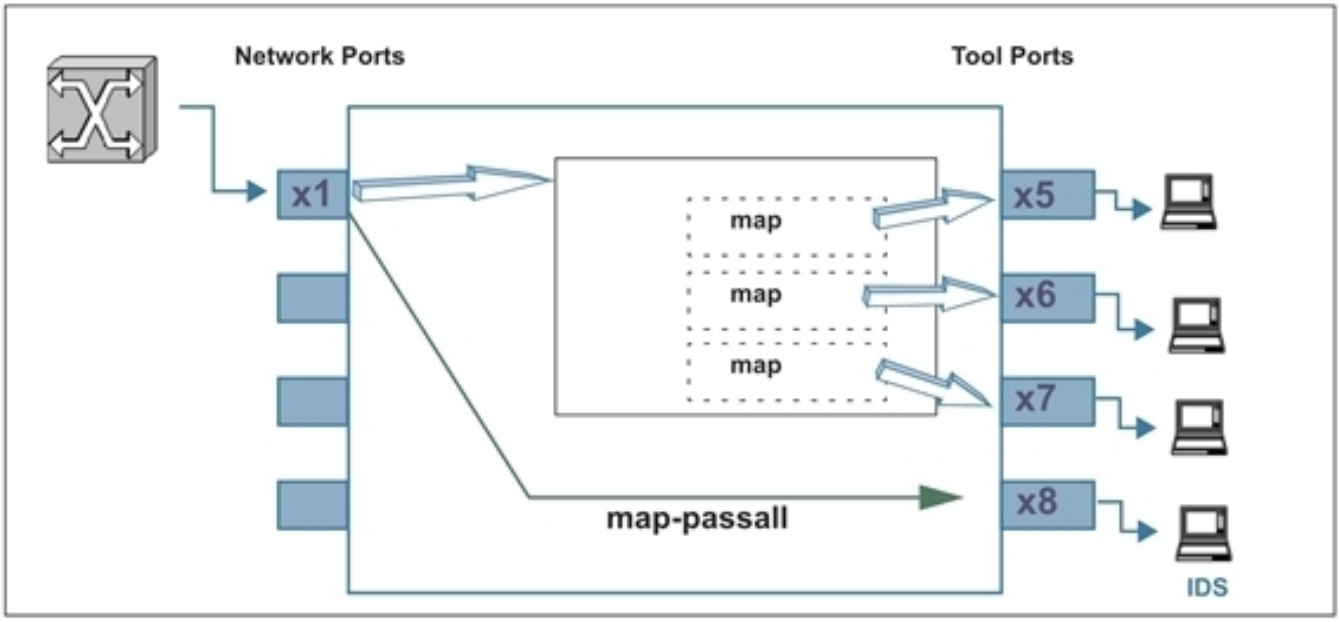


Figure 5 *Sending All Traffic to IDS*

Temporary Troubleshooting Situations

Under certain circumstances, you may want to see all of the traffic on a particular port without disturbing any of the packet distribution commands already in place for the port. The **map-passall** gives you a way to do this. For example, suppose you have existing maps sending traffic from network port 1/2/x1 to tool ports 1/2/x5..x7 based on different map rule criteria (Figure 6Existing Map on Network Port 1/2/x1).

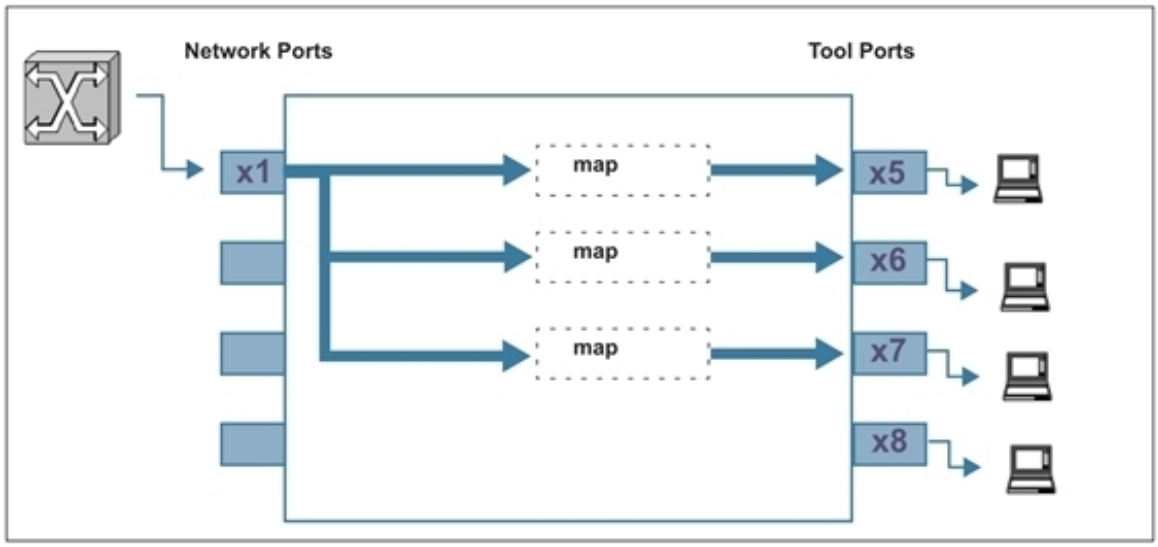


Figure 6 *Existing Map on Network Port 1/2/x1*

Complaints of slow response times on the network monitored by network port 1/2/x1 lead you to want to see **all** of the traffic rather than just the portions broken out by your maps. Because a packet goes only to the destination specified by the map with the highest priority, you cannot just create a new map with no rules to see all of the traffic on the port. However, you also do not want to take down your existing maps.

In a situation like this, you could set up a **map-passall** for the mapped network port and send the full set of traffic arriving at the network port to another tool port. For example:

```
(config) # map-passall alias temp_pass (config map temp_pass) # from 1/2/x1 (config map temp_pass) # to 1/2/x8
(config map temp_pass) # exit
```

Now, the full set of traffic arriving on network port 1/2/x1 is both passed to tool port 1/2/x8 and also distributed to network ports 1/2/x5..x7 based on the existing maps (Figure 7Adding a Map-passall for Temporary Troubleshooting).

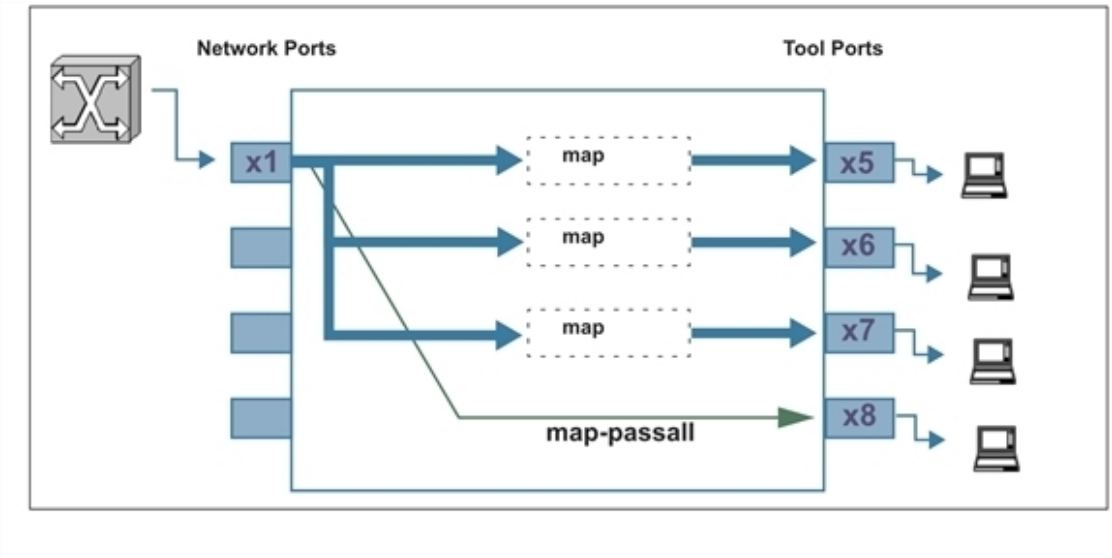


Figure 7 *Adding a Map-passall for Temporary Troubleshooting*

Example: How to work with Multiple Rules in the Same Map

Rules created in the same map work as "**AND**". This means, the Rule1 logic will use the cumulative "AND" operator to include Rule 2 and will only apply the map to the egress port when all conditions are met.

```
map alias mapallrules
type regular byRule
roles replace admin to owner_roles
rule add pass vlan 924
rule add pass vlan 3009
rule add pass vlan 3014
```

```
rule add pass vlan 3017
rule add pass vlan 3020
rule add drop portsrc 1556 protocol tcp
rule add drop portdst 1556 protocol tcp
to 1/1/x6
from 1/1/x2
exit
```

In this example, the expected output would be that if packets coming in through port x2 from vlan 924 that comes from port src ID 1556 then, drop rule does apply and packets get dropped even though the pass rule was in place for vlan 924.

How to Send Tool-Port Filtered Traffic to Multiple Destinations: tool-mirror

You can use the tool-mirror command to see the same tool-port-filtered data on multiple tool ports.

Consider the following scenario:

- Network ports 1/3/x1..x3 have maps sending different data to tool port 1/3/x5.
- Tool port 1/3/x5 has a port-filter set up to allow only VLAN IDs 100-500.

Figure 8Creating a Tool-mirror illustrates this scenario.

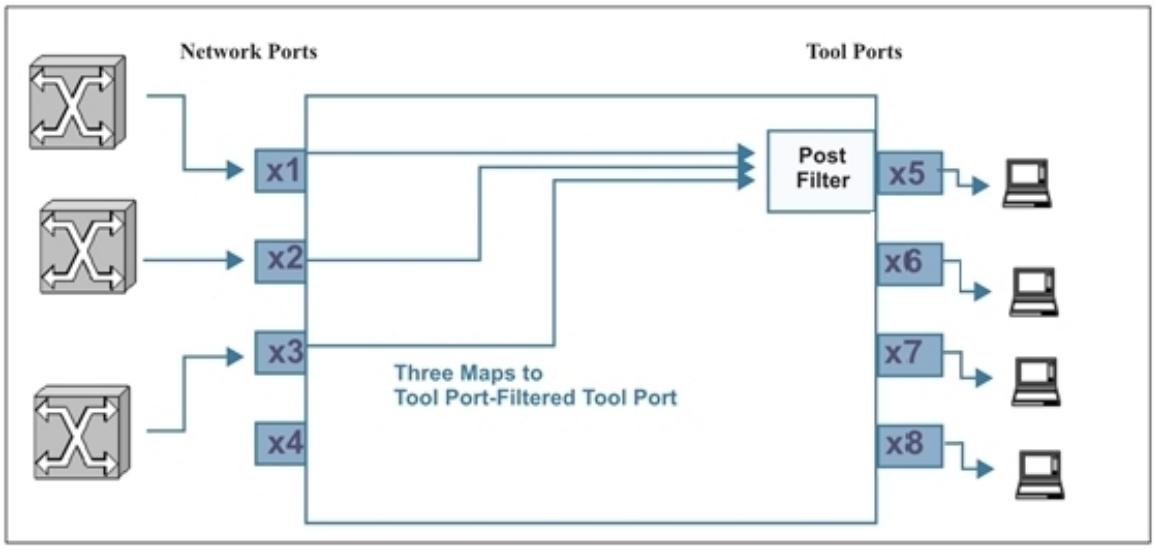


Figure 8 *Creating a Tool-mirror*

If you wanted different tools to analyze the same tool-port-filtered data, you could set up a tool-mirror to multiple tool ports so that they could all see the same data. For example:

```
(config) # tool-mirror alias toolplus from 1/2/x5 to 1/2/x6..x8
```

With this configuration (Figure 9Adding tool-mirrors to Multiple Tool Ports), tool ports 1/2/x5 to x8 all see the same tool-port-filtered data.

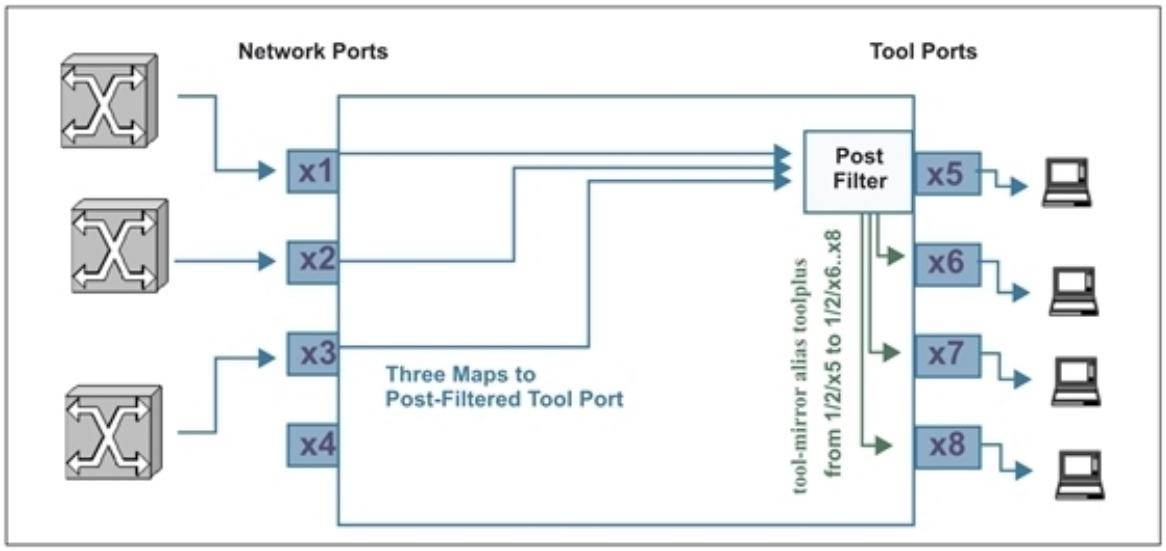


Figure 9 *Adding tool-mirrors to Multiple Tool Ports*

Example of Hybrid Ports

In this example, the hybrid ports duplicate traffic from one network source after removing the MPLS header.

Step	Description	Command
1.	Configure ports.	**(config) # port 17/1/x1 type network** **(config) # port 17/1/x2..x3 type hybrid** **(config) # port 17/1/x5..x6 type tool**
2.	Create a GigaSMART group and specify the GigaSMART engine port.	**(config) # gsgroup alias group17_5 port-list 17/5/e1**
3.	Create a GSOP, including the application and the GigaSMART group.	**(config) # gsop alias gsop1 mpls-header-strip strip-header mpls port-list group17_5**
4.	Create a map to remove the MPLS header.	**(config) # map alias map1** **(config map alias map1) # type regular byRule** **(config map alias map1) # roles replace admin to owner_roles** **(config map alias map1) # use gsop mpls-header-strip** **(config map alias map1) # rule add pass macsrc**

Step	Description	Command
		0000.0000.0000 0000.0000.0000 **(config map alias map1) # to 17/1/x2..x3** **(config map alias map1) # from 17/1/x1** **(config map alias map1) # exit** **(config) #**
5.	Create a second map.	**(config) # map alias map2** **(config map alias map2) # type regular byRule** **(config map alias map2) # roles replace admin to owner_roles** **(config map alias map2) # rule add pass ipsrc 10.120.7.12 255.255.255.0** **(config map alias map2) # rule add drop ipdst 192.168.0.100 255.255.255.255** **(config map alias map2) # rule add drop ipdst 192.168.0.101 255.255.255.255** **(config map alias map2) # rule add drop ipdst 192.168.0.102 255.255.255.255** **(config map alias map2) # rule add drop ipdst 192.168.0.103 255.255.255.255** **(config map alias map2) # to 17/1/x5** **(config map alias map2) # from 17/1/x2** **(config map alias map2) # exit** **(config) #**
6.	Create a third map.	**(config) # map alias map3** **(config map alias map3) # type regular byRule** **(config map alias map3) # roles replace admin to owner_roles** **(config map alias map3) # rule add pass ipdst 192.168.0.1 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.10 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.100 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.101 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.102 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.103 255.255.255.255 bidir** **(config map alias map3) # rule add pass ipdst 192.168.0.104 255.255.255.255 bidir**

Step	Description	Command
		(config map alias map3) # rule add pass ipdst 192.168.0.105 255.255.255.255 bidir **(config map alias map3) # to 17/1/x6** **(config map alias map3) # from 17/1/x3** **(config map alias map3) # exit** **(config) #**
7.	Display the configuration.	show gsgroupshow gsopshow map

Use the following command to display the GigaSMART group configuration:

(config) # show gsgroup

Port-Filter Examples

The following table provides some examples of egress port-filters:

Description	Command
The following egress port-filter drops all packets with a VLAN ID between 100..200 from tool port 14/2/g40:	**(config) # port 14/2/g40 filter rule add drop vlan 100..200**
Similarly, this command passes only IPv6 traffic on egress port 14/2/g44:	**(config) # port 14/2/g44 filter rule add pass ipver 6**

Configure Active Visibility

Active Visibility is a framework that allows your visibility network to adapt to dynamic events. The framework is designed to react to events and take actions in response to events in your visibility network.

An Active Visibility policy defines conditions and actions. When conditions are met, actions are executed. The policy specifies both the conditions and the actions and ties them together.

The configuration examples for Active Visibility is described in the following sections:

- Conditions
- Actions
- Policies

Related Topics

- Refer to the *"Configure Active Visibility"* section in the GigaVUE Fabric Management Guide for detailed information.
- Refer to the policy in the reference section for details on the syntax of the policy CLI command.

Conditions

Conditions in a policy are events that can trigger actions. Use the **policy** command with **condition add** to define the conditions in a policy. For example:

```
(config) # policy alias MapMonitor condition add PortDown param portId 2/3/q2
```

For example, refer to the template for the **PortDown** condition as follows:

```
The link of port <$portId$> is down for a period of [$period$] second(s).
```

In the example, the mandatory PortId keyword has been specified with a value of 2/3/q2.

There is also an optional period keyword that can be specified in the condition. For example:

```
(config) # policy alias MapMonitor condition add PortDown param portId 2/3/q2 param period
120
```

These examples define a port down condition for port 2/3/q2. When used in a policy, the condition in the first example will be met when the port is down. In the second example, the condition will be met when the port is down for a period of 120 seconds.

How to Specify Keywords in Conditions

The syntax for specifying keywords in conditions is described in the following sections:

Specifying Keyword, gsGroup

The mandatory keyword, gsGroup, specifies a GigaSMART group created.

For Example:

To detect the High CPU utilization for a GigaSMART group, create a policy as follows:

```
(config) # policy alias GSPolicies condition add GsCpuUtilHigh param gsGroup testgroup
param threshPct 90 param period 120
```

Specifying Keyword, inlineToolAlias

The mandatory keyword, inlineToolAlias, specifies an inline tool configured, for example:

(config) # policy alias p1 condition add InlineToolReady param inlineToolAlias testinlinetool

Specifying Keyword, inlineToolGrpAlias

The mandatory keyword, inlineToolGrpAlias, specifies an inline tool group configured, for example:

(config) # policy alias p1 condition add InlineToolGrpDn param inlineToolGrpAlias itg1

Specifying Keyword, period

The optional keyword, period, specifies a time in seconds, for example:

(config) # policy alias MapMonitor condition add PortDown param portId 2/3/q2 param period 120

Refer to the template for the **PortDown** condition as follows:

The link of port <$portId$> is down for a period of [$period$] second(s).

Period means wait for the number of seconds specified. During that waiting period, the condition might be met, for example, if the port stays down for 120 seconds. But if the condition is not met (for example, the port comes back up before 120 seconds have passed), then the condition will not be met. If the port goes down again, the 120 seconds waiting period will start again.

When a period is specified, it provides a dampening or soaking period. For example, you might not want to execute an action unless a threshold has been exceeded for a certain amount of time.

If you are monitoring ports, they might flap. It is recommended that you always specify a period when using **PortUp** and **PortDown** conditions.

Specifying Keyword, thresh

The mandatory keyword, thresh, specifies a threshold value, for example:

(config) # policy alias PortPolicy condition add PortRxDiscardsHigh param portId 2/3/x1 param thresh 100

Refer to the template for the **PortRxDiscardsHigh** condition as follows:

The port <$portId$> Rx discards is greater than <$thresh$> for a period of [$period$] second (s).

Use high and low thresholds to define a range of values.

NOTE: For any discard, drop, or error condition, the values can only go up because they are counters that start at zero and are then incremented. Note that the counters for any discard, drop, or error condition can be cleared.

To clear counters, use the following command:

```
(config) # clear port stats
```

Specifying Keyword, threshPct

The mandatory keyword, threshPct, specifies a threshold percentage, for example:

```
(config) # policy alias OverloadedToolPort condition add PortTxUtilHigh param portId 1/1/x1 param threshPct 80
```

In this example, the utilization needs to be higher than the threshold percentage of 80 for the condition to be met.

Refer to the template for the **PortTxUtilHigh** condition as follows:

```
The port <$portId$> Tx utilization is greater than <$threshPct$> for a period of [$period$] second(s).
```

The optional keyword, period, can be added to the condition as follows:

```
(config) # policy alias OverloadedToolPort condition add PortTxUtilHigh param portId 1/1/x1 param threshPct 80 param period 120
```

In this example, the utilization needs to be higher than the threshold percentage of 80 for a period of 120 seconds. The condition will be met if the threshold is above 80 for 120 seconds. However, during the 120 seconds, if the threshold drops below 80, the condition will not be met. If the threshold goes over 80 again, a new 120 seconds waiting period will start.

Specifying Keyword, timeStr

The mandatory keyword, timeStr, specifies a time string, for example:

```
(config) # policy alias SaveMemory condition add TimeOfDay param timeStr "( 45 10 * * * * )"
```

Refer to the template for the **TimeOfDay** condition as follows:

```
Time of day is <$timeStr$>
```

The format of the timeStr keyword is a Cron format, which has the form: "(a b c d e f)".

Table 2: Cron Format lists the six fields in the Cron format.

Table 2: Cron Format

Key	Description	Range	Notes
a	**minute**	0-59	
b	**hour**	0-23	
c	**day of the month**	1-31	
d	**month of the year**	1-12	or Jan, Feb...Dec
e	day of the week	1-7	where 1 = Monday, or Mon, Tue...Sun
f	**year**	1900-3000	

In addition to the numbers listed in the Range column, each field supports a wildcard (*) character.

After a wildcard or a range of values, you can use the slash (/) to specify values that are repeated over and over, with an interval in between.

Only the Cron formats listed above are supported.

NOTE: When a time string is specified, such as for one minute or for one hour, it means that the condition will be matched every minute or every hour. This can result in a condition that triggers an action again, even if it was already triggered.

Time-based conditions are evaluated at the point at which the time changes. For example, if the condition is **TimeWednesday**, it is evaluated when Tuesday changes to Wednesday, which occurs just after midnight, in the first minute of Wednesday.

Specifying Keyword, portId

The mandatory keyword, portId, specifies a port identifier. There are several ways to specify ports. Whether or not a condition is met depends on how the ports are specified.

The formats for portId are as follows:

- single port—a/b/c
- multiple ports, separated by commas—a1/b1/c1,a2/b2/c2
- range of ports—a/b/c..d
- any port—any(a/b/c..d), which includes the keyword, any.

The following is an example of a single port. When the port matches, the condition is met.

(config) # policy alias SingleUpPort condition add PortUp param portId 1/1/x1

Refer to the template for the **PortUp** condition as follows:

The link of port <$portId$> is up for a period of [$period$] second(s).

The following is an example of a range of ports. When all ports match, the condition is met. In this case, all ports means x1, x2, and x3.

(config) # policy alias RangeUpPort condition add PortUp param portId 1/1/x1..x3

The following is an example of any ports. When any port matches, the condition is met. In this case, any port means x1, x2, or x3.

(config) # policy alias AnyUpPort condition add PortUp param portId any(1/1/x1..x3)

The optional keyword, period, can be added to the condition as follows:

(config) # policy alias TwoUpPorts condition add PortUp param portId 1/1/x1..x2 param period 120

In this example assume that x1 comes up and then x2 comes up. When both ports are up, wait 120 seconds. At this point, the condition is met and the trigger action starts.

If the condition was specified as follows using the keyword any, the condition will be met after either port or both ports come up and stay up for 120 seconds:

(config) # policy alias TwoUpPorts condition add PortUp param portId any(1/1/x1..x2) param period 120

Actions

Actions in a policy can notify users of certain events or change the configuration in response to events.

Use the **policy** command with **action add** to define the actions in a policy. For example:

(config) # policy alias MapMonitor action add MapDisable param mapAlias map1

Refer to the template for the **MapDisable** action as follows:

Disabling map <$mapAlias$>.

In the example, the mandatory **mapAlias** keyword has been specified. Some actions have multiple mandatory keywords.

Use the **show action** command to view the pre-defined actions and their templates. The command is as follows:

(config) # show action

In the output of the **show action** command, the Template column displays the parameters that must be specified when defining the action in a policy. The template contains mandatory keywords, for example, <$mapAlias$>. The strings enclosed in dollar signs ($) are parameters that must be specified as part of configuring a policy.

NOTE: Some actions do not have any parameters, such as **WriteMemory**.

How to specify Keywords in Actions

The syntax for specifying keywords in actions is described in the following sections:

Specifying Keyword, mapAlias

The mandatory keyword, mapAlias, specifies a map alias, for example:

```
(config) # policy alias p1 action add MapDisable param mapAlias m1
```

Refer to the template for the **MapDisable** action as follows:

```
Disabling map <$mapAlias$>
```

Specifying Keyword, policyAlias

The mandatory keyword, policyAlias, specifies a policy alias, for example:

```
(config) # policy alias p1 action add PolicyEnable param policyAlias p1
```

Refer to the template for the **PolicyEnable** action as follows:

```
Enabling policy <$policyAlias$>
```

Specifying Keyword, portId

The mandatory keyword, portId, specifies a port identifier. There are several ways to specify ports.

The formats for portId are as follows:

- single port—a/b/c
- multiple ports, separated by commas—a1/b1/c1,a2/b2/c2
- range of ports—a/b/c..d

The following is an example of a single port. When the port matches, the action is executed.

```
(config) # policy alias policy1 action add PortEnable param portId 1/1/x1
```

Refer to the template for the **PortEnable** action as follows:

Enabling port <$portId$> admin.

The following is an example of multiple ports. When all ports match, the action is executed. In this case, all ports means x1 and x3.

(config) # policy alias policy3 action add PortEnable param portId 1/1/x1,1/1x3

Specifying Keyword, ruleId

The mandatory keyword, ruleId, specifies a map rule identifier, for example:

(config) # policy alias policy2 action add MapRuleDelete param ruleId 2 param mapAlias map1

In this example, if there were five rule IDs numbered from 1 to 5 and rule ID 2 is removed, the remaining rule IDs will be numbered 1, 3, 4, and 5.

Refer to the template for the **MapRuleDelete** action as follows:

Remove rule-id <$ruleId$> from map <$mapAlias$>

To obtain rule IDs, use the **show map** command.

Specifying Keyword, ruleStr

The mandatory keyword, ruleStr, specifies a map rule string, for example:

(config) # policy alias AddM1 action add MapRuleAdd param ruleStr "pass vlan 100" param mapAlias m1

Refer to the template for the **MapRuleAdd** action as follows:

Add a rule <$ruleStr$> to map <$mapAlias$>

Rule strings begin with **pass** or **drop**. The rule strings are enclosed in quotation marks.

Specifying Keyword, inlineNetAlias

The mandatory keyword, inlineNetAlias, specifies an inline network port alias, for example:

(config) # policy alias p1 action add PhysicalByPassEnable param inlineNetAlias IN1

Refer to the template for the **PhysicalByPassEnable** action as follows:

Enable Physical Bypass for Inline Network Port <$inlineNetAlias$>

Specifying Keyword, inlineToolAlias

The mandatory keyword, inlineToolAlias, specifies an inline tool port alias, for example:

(config) # policy alias p1 action add InlineToolEnable param inlineToolAlias IT1

Refer to the template for the **InlineToolEnable** action as follows:

Enabling inline tool <$inlineToolAlias$>

Specifying Keyword, inlineNetTrafficPath

The mandatory keyword, inlineNetTrafficPath, specifies the traffic path for the inline network, for example:

(config) # policy alias p1 action add InlineNetTrafficPath param inlineNetTrafficPath bypass param inlineNetAlias IN1

Refer to the template for the **InlineNetTrafficPath** action as follows:

Set Traffic-Path as <$inlineNetTrafficPath$> for Inline network <$inlineNetAlias$>

Specifying Keyword, oobFromAlias

The mandatory keyword, oobFromAlias, specifies an inline network alias to add or remove out-of-band copy for flexinline map, for example:

(config) # policy alias AddM1 action add FlexInlineOOBDelete param mapAlias flexmap1 param oobFromAlias IN1

Refer to the template for the **FlexInlineOOBDelete** action as follows:

Remove OOBCopy for map <$mapAlias$> from Inline network <$oobFromAlias$>

Specifying Keyword, oobDir

The mandatory keyword, oobDir, specifies the direction of traffic when you add or remove out-of-band copy for flexinline map, for example:

(config) # policy alias AddM1 action add FlexInlineOOBAddWithDir param mapAlias flexmap1 param oobFromAlias IN1 param oobDir a-to-b param portId 20/3/q4

Refer to the template for the **FlexInlineOOBAddWithDir** action as follows:

Create OOBCopy for map <$mapAlias$> from <$oobFromAlias$> for direction <$oobDir$> to OOBTool <$portId$>

Specifying Keyword, oobTag

The mandatory keyword, oobTag, specifies the tag for out-of-band tool port, for example:

(config) # policy alias AddM1 action add FlexInlineOOBAddWithDirTag param mapAlias flexmap1 param oobFromAlias IN1 param oobDir a-to-b param oobTag as-inline param portId 20/3/q5

Refer to the template for the **FlexInlineOOBAddWithDirTag** action as follows:

Create OOBCopy for map <$mapAlias$> from <$oobFromAlias$> for direction <$oobDir$> to OOBTool <$portId$> with tag <$oobTag$>

Related Topics

Also, refer to the following sections:

Map Enable and Disable

Refer to map.

Known Behaviors

The following are known behaviors:

- Events are evaluated sequentially. Actions are executed sequentially.
- If an unsupported map type is specified in an action, the output of the **show policy** command displays Action error!
- If an alias or port ID does not exist, the following error is displayed: Invalid param value.

Actions in Policies

For the usage of actions in policies, refer to Policies.

Policies

Policies tie conditions and actions together. Enable the policy for it to take effect.

Refer to the following sections for configuration examples:

- GigaSMART Group Policy
- Inlinetool Policy
- Overloaded Tool Port Policy
- Weekend Policy
- Any Port Up Policy
- All Ports Up Policy
- Map Disable Policy
- Redundant Map Policy

- Revert a Redundant Map Policy
- Save Memory Policy
- High Availability Policy
- Tool Optimization Policy
- Automated Monitoring Policy
- Enable Map Based on Time Policy

Refer also to the following sections:

- Parameter Passing
- How to Edit Policies

GigaSMART Group Policy

Use the following steps to configure a GigaSMART group policy. You must configure the **gsgroup** and the **map1** before configuring GigaSMART Group policy.

```
(config) # policy alias GsGroupPolicy
(config policy alias p1) # condition add GsCpuUtilHigh param gsGroup gsgroup1
param threshPct 50
(config policy alias p1) # enable
(config policy alias p1) # action add MapGsRuleDelete param mapAlias map1 param
ruleId 1
(config policy alias p1) # exit
```

Inlinetool Policy

Use the following steps to configure an Inlinetool policy. You must configure map1 and testinlinetool before configuring Inlinetool policy.

(config) # policy alias p1 (config policy alias p1) # condition add InlineToolReady param inlineToolAlias testinlinetool (config policy alias p1) # enable (config policy alias p1) # action add MapEnable param mapAlias map1 (config policy alias p1) # exit

Overloaded Tool Port Policy

Use the following steps to configure an overloaded tool port policy:

(config) # policy alias OverloadedToolPort condition add PortTxUtilHigh param portId 1/1/x1 param threshPct 80 (config) # policy alias OverloadedToolPort action add PortDisable param portId 1/1/x2 (config) # policy alias OverloadedToolPort enable

The following is the same example for an overloaded tool port, but using the prefix mode:

(config) # policy alias OverloadedToolPort

(config policy alias OverloadedToolPort) # condition add PortTxUtilHigh param portId 1/1/x1 param threshPct 80

(config policy alias OverloadedToolPort) # action add PortDisable param portId 1/1/x2 (config policy alias OverloadedToolPort) # enable

(config policy alias OverloadedToolPort) # exit

(config) #

Weekend Policy

Use the following steps to configure a weekend policy. The map, WeekendMap, would also have to be configured.

To enable a weekend map on the weekend:

(config) # policy alias WeekendEn (config policy alias WeekendEn) # condition add TimeWeekend (config policy alias WeekendEn) # action add MapEnable param mapAlias WeekendMap (config policy alias WeekendEn) # enable (config policy alias WeekendEn) # comment “Enable WeekendMap on the weekend” (config policy alias Weekend) # exit (config) #

To disable the weekend map on weekdays:

(config) # policy alias WeekendDis (config policy alias WeekendDis) # condition add TimeWeekday (config policy alias WeekendDis) # action add MapDisable param mapAlias WeekendMap (config policy alias WeekendDis) # enable (config policy alias WeekendDis) # comment “Disable WeekendMap on weekdays” (config policy alias WeekendDis) # exit (config) #

Any Port Up Policy

Use the following steps to configure a policy for any port up:

(config) # policy alias AnyPortUp (config policy alias AnyPortUp) # condition add PortUp param portId any(3/1/q4..q6) param period 300 (config policy alias AnyPortUp) # action add PortFilterAdd param portId &PortUp.portId& param ruleStr "add pass vlan 100" (config policy alias AnyPortUp) # enable (config policy alias AnyPortUp) # exit (config) #

This example uses parameter passing. Refer to Parameter Passing for details.

All Ports Up Policy

Use the following steps to configure a policy for all ports up:

(config) # policy alias AllPortUp

(config policy alias AllPortUp) # condition add PortUp param portId any(3/1/q4..q6) param period 300

(config policy alias AllPortUp) # action add PortFilterAdd param portId &PortUp.portId& param ruleStr "add pass vlan 100"

(config policy alias AllPortUp) # enable

(config policy alias AllPortUp) # exit

(config) #

Map Disable Policy

Use the following steps to configure a policy for disabling a map when a tool port is down:

Create the map and enable it:

```
(config) # map alias map1
(config map alias map1) # from 2/3/g1
(config map alias map1) # to 2/3/g2
(config map alias map1) # rule add pass vlan 100
(config map alias map1) # enable
(config map alias map1) # exit
(config) #
```

Create the policy and enable it:

```
(config) # policy alias MapMonitor
(config policy alias MapMonitor) # condition add PortDown param portId 2/3/g2
(config policy alias MapMonitor) # action add MapDisable param mapAlias map1
(config policy alias MapMonitor) # enable
(config policy alias MapMonitor) # exit
(config) #
```

Redundant Map Policy

Use the following steps to configure a policy for a redundant map. This policy has multiple actions.

```
(config) # policy alias RedundantMap
(config policy alias RedundantMap) # condition add PortDown param portId 2/2/c1
(config policy alias RedundantMap) # action add MapEnable param mapAlias map2
(config policy alias RedundantMap) # action add MapDisable param mapAlias map1
(config policy alias RedundantMap) # enable
(config policy alias RedundantMap) # exit
(config) #
```

Revert a Redundant Map Policy

Use the following steps to configure a policy for reverting a redundant map. This policy also has multiple actions.

```
(config) # policy alias RevertRedundantMap
(config policy alias RevertRedundantMap) # condition add PortUp param portId 2/2/c1
(config policy alias RevertRedundantMap) # action add MapEnable param mapAlias map1
(config policy alias RevertRedundantMap) # action add MapDisable param mapAlias map2
(config policy alias RevertRedundantMap) # enable
(config policy alias RevertRedundantMap) # exit
(config) #
```

Save Memory Policy

Use the following steps to configure a policy for saving memory:

```
(config) # policy alias SaveMemory
(config policy alias SaveMemory) # condition add TimeOfDay param timeStr “( 45 10 * * * * )”
(config policy alias SaveMemory) # action add WriteMemory
(config policy alias SaveMemory) # enable
(config policy alias SaveMemory) # exit
(config) #
```

High Availability Policy

Use the following steps to configure policies for high availability. In this example, there are two maps, each to a different tool port, for example, tool1 and tool2. The policies define that if tool1 is down, use tool2. If tool1 comes back up, switch back to it.

Configure two maps:

```
(config) # map alias map1
(config map alias map1) # from 1/1/x1
(config map alias map1) # to 1/1/x11
(config map alias map1) # rule add pass vlan 100
(config map alias map1) # enable
(config map alias map1) # exit
(config) #
(config) # map alias map2
(config map alias map2) # from 1/1/x1
(config map alias map2) # to 1/1/x12
(config map alias map2) # rule add pass vlan 100
(config map alias map2) # no enable
(config map alias map2) # exit
(config) #
```

Configure two policies, each with multiple actions:

```
(config) # policy alias HA1
(config policy alias HA1) # condition add PortDown param portId 1/1/x11
(config policy alias HA1) # action add MapEnable param mapAlias map2
(config policy alias HA1) # action add MapDisable param mapAlias map1
(config policy alias HA1) # action add PolicyEnable param policyAlias HA2
(config policy alias HA1) # enable
(config policy alias HA1) # exit
(config) #
 (config) # policy alias HA2
(config policy alias HA2) # condition add PortUp param portId 1/1/x11
```

```
(config policy alias HA2) # action add MapDisable param mapAlias map2
(config policy alias HA2) # action add MapEnable param mapAlias map1
(config policy alias HA2) # enable
(config policy alias HA2) # exit
(config) #
```

Tool Optimization Policy

Use the following steps to configure policies for tool optimization.

Configure a map:

```
(config) # map alias map1
(config map alias map1) # from 1/1/x3
(config map alias map1) # to 1/1/x5
(config map alias map1) # rule add pass vlan 200
(config map alias map1) # rule add pass vlan 100
(config map alias map1) # enable
(config map alias map1) # exit
(config) #
```

Configure two policies:

```
(config) # policy alias ToolOpt1
(config policy alias ToolOpt1) # condition add PortRxUtilHigh param portId 1/1/x3 param threshPct 80
(config policy alias ToolOpt1) # action add MapRuleDelete param mapAlias map1 param ruleId 2
(config policy alias ToolOpt1) # action add PolicyEnable param policyAlias ToolOpt2
(config policy alias ToolOpt1) # enable
(config policy alias ToolOpt1) # exit
(config) #
 (config) # policy alias ToolOpt2
(config policy alias ToolOpt2) # condition add PortRxUtilLow param portId 1/1/x3 param threshPct 81
(config policy alias ToolOpt2) # action add MapRuleAdd param mapAlias map1 param ruleStr “pass vlan 100”
(config policy alias ToolOpt2) # action add PolicyEnable param policyAlias ToolOpt1
(config policy alias ToolOpt2) # no enable
(config policy alias ToolOpt2) # exit
(config) #
```

Automated Monitoring Policy

Use the following steps to configure a policy for automated monitoring. In this example, if the source is 1.1.1.1, enable the map.

Configure a map:

```
(config) # map alias map1
(config map alias map1) # from 1/1/x1
(config map alias map1) # to 1/1/x2
(config map alias map1) # no enable
(config map alias map1) # exit
(config) #
```

Configure the policy:

```
(config) # policy alias AutoMon
(config policy alias AutoMon) # condition add PortRxUtilHigh param portId 1/1/x1 param threshPct 30
(config policy alias AutoMon) # action add MapEnable param mapAlias map1
(config policy alias AutoMon) # enable
(config policy alias AutoMon) # exit
(config) #
```

Enable Map Based on Time Policy

Use the following steps to configure policies for enabling a map based on time, such as between the hours of 8:00am and 5:00pm (17:00). During the other hours (from midnight to 8:00am and from 5:00pm to midnight), disable the map.

```
(config) # policy alias WorkHours
(config policy alias WorkHours) # condition add TimeOfDay param timeStr “( 0 8-17 * * * * )”
(config policy alias WorkHours) # action add MapEnable param mapAlias map1
(config policy alias WorkHours) # enable
(config policy alias WorkHours) # comment “Enable map1 from 8:00 to 17:00 daily”
(config policy alias WorkHours) # exit
(config) # (config) # policy alias MidnightHours
(config policy alias MidnightHours) # condition add TimeOfDay param timeStr “( 0 0-8 * * * * )”
(config policy alias MidnightHours) # action add MapDisable param mapAlias map1
(config policy alias MidnightHours) # enable
(config policy alias MidnightHours) # comment “Disable map1 from midnight to 8:00 daily”
(config policy alias MidnightHours) # exit
(config) # (config) # policy alias AfterWorkHours
(config policy alias AfterWorkHours) # condition add TimeOfDay param timeStr “( 0 17-0 * * * * )”
(config policy alias AfterWorkHours) # action add MapDisable param mapAlias map1
(config policy alias AfterWorkHours) # enable
(config policy alias AfterWorkHours) # comment “Disable map1 from 17:00 to midnight daily”
(config policy alias AfterWorkHours) # exit
(config) #
```

Parameter Passing

Parameters can be passed from a condition to an action in a policy. Parameter passing is currently only for ports.

For example:

```
(config) # policy alias PortPolicy
(config policy alias PortPolicy) # condition add PortUp param portId any(1/1/x1..x3)
(config policy alias PortPolicy) # action add PortDisable param portId &PortUp.portId&
(config policy alias PortPolicy) # enable
(config policy alias PortPolicy) # exit
(config) #
```

Parameter passing is specified using two ampersand (&) symbols, for example, &PortUp.portId& in the action. The **PortUp** and the **portId** in the action match the **PortUp** and **portId** in the condition, as specified between the two ampersands.

The value for the parameter will be substituted and only the condition that was met will be substituted.

In this example, the condition is for any port in the range of 1/1/x1..x3 to be up. If only port x3 is up (matching the condition), only x3 will be passed to the action. If port x1 and x2 are both up at the same time, both will be passed to the action.

If the condition had been specified without the keyword, any, as follows:

```
(config policy alias PortPolicy) # condition add PortUp param portId 1/1/x1..x3
```

The ports x1, x2, and x3 would all have to be up at the same time to match the condition and be passed to the action.

NOTE: Use caution with parameter passing. When a parameter is passed from a condition to an action, the system does not validate the parameter that is passed. For example, you can configure a policy that attempts to pass a TimeofDay parameter, timeStr in a condition to a PortEnable action. But parameter passing is currently only supported for ports, so timeStr is not a valid parameter to pass. If the policy is triggered, the action will not be successful.

How to Edit Policies

To edit a policy, it is recommended that you disable the policy first. To disable a policy, use the following line-by-line command:

```
(config) # no policy alias policy1 enable
```

Using the prefix mode is not recommended The changes only take effect when you exit. The following is an example of the prefix mode:

```
(config) # policy alias policy1
(config policy alias policy1) # no enable
(config policy alias policy1) # exit
```

Edit the policy. The following edits are supported:

- Add a new condition or a new action to a policy, so long as the limits of 5 conditions and 5 actions in a policy are not exceeded.
- Delete an existing condition or an existing action from a policy.
- Modify a condition only by deleting it and adding it back in with the change. An existing condition cannot be modified by editing it.

NOTE: An action or a condition is deleted by its ID (action ID or condition ID). These IDs are incremented. For example, if three conditions are defined, they will be numbered 1, 2, and 3. If 2 is deleted, the other IDs will be numbered 1, 3. Once 2 is deleted, it is no longer used. If a condition is added, it will take the next available number, in this case 4.

Finally, re-enable the policy with the following command:

```
(config) # policy alias policy1 enable
```

Note that editing a policy automatically resets the policy, meaning the run counts are reset to zero. The run counts and action status are reset when the node reboots or when there are cluster changes, such as a leader switchover.

Configure Custom Interface Selection

When the interface IE is not available in the PFCP packets, you can explicitly provide the IP addresses of the network nodes such as SGW-C and SGW-U for identifying and filtering the traffic based on the interface (S1U, S5S8-U and N3). Refer to Interface Filtering and APN/DNN Filtering for more details.

Creation of IP Profile

upn-interface-profile alias <name>

profile add type sgwc ipv4 <ip-address or ip-address range with mask>

profile add type s5s8u ipv4 <ip-address or ip-address range with mask>

exit

Deletion of IP Profile

no upn-interface-profile alias <name>

NOTE: You cannot delete an IP profile when it is associated with a gsgroup.

You will get the following error message when you try to delete a IP profile that is associated with a gsgroup.

```
gigamon-330312 [SA-UPN: leader] (config) # no upn-interface-profile alias test
% UPN Interface profile test is in use, cannot remove.
```

Gsparams

gsparams gsgroup <alias>

upn-interface-select type <3gpp | custom> [profile <upn-profile-name>]

exit

Disassociation of IP Profile from a gsgroup

You need to choose **3gpp** role using the following command to disassociate an IP profile from a gsgroup.

upn-interface-select type 3gpp

Configure GigaStream

A GigaStream groups multiple ports into a logical bundle. Use the **gigastream** command to configure a GigaStream. There are two types of GigaStream: regular GigaStream and controlled GigaStream. Both types of GigaStream bundle multiple ports to provide logical bandwidth. Packets arriving through network ports are processed with various map rules and then directed to ports. All traffic streams destined to a GigaStream are hashed among the bundled ports.

The configuration examples for configuring GigaStream are described in the following sections:

- Regular GigaStream Configuration
- Controlled GigaStream Configuration
- Advanced Hashing
- Weighted GigaStream

Related Topics

- Refer to the "*GigaStream*" section in the *GigaVUE Fabric Management Guide* for detailed information.

- Refer to the gigastream in the reference section for details on the syntax of the GigaStream CLI command.
- Refer to the gigastream advanced-hash in the reference section for details on the syntax of the GigaStream advanced hash CLI command.

Regular GigaStream Configuration

To configure a regular tool GigaStream, refer to the following example:

Step	Description	Command
1.	Configure ports using type tool for a regular tool GigaStream.	**(config) # port 1/3/q2..q3 type tool**
2.	Configure a regular GigaStream.	**(config) # gigastream alias stream1 port-list 1/3/q1..q4**
3.	Configure a comment for the GigaStream.	**(config) # gigastream alias stream1 comment “regular gigastream”**
4.	Assign hash weights in percentage or ratio to the ports in the GigaStream	**1/3/q1..q4 hash-weight 30,30,20,20** **1/3/q1..q4 hash-weight 3,3,2,2**
5.	Assign drop weight for the GigaStream	(config gigastream alias stream1) # drop-weight 2
6.	Display the configuration for this example.	**(config) # show gigastream**

Controlled GigaStream Configuration

To configure a controlled tool GigaStream, specify hash size and hash bucket ID, using the prefix mode. Refer to the following example:

Step	Description	Command
1.	Configure ports using type tool for controlled GigaStream.	**(config) # port 1/3/q4..q6 type tool**
2.	Configure a controlled GigaStream. This uses the prefix mode to configure all parameters.	**(config) # gigastream alias stream2** **(config gigastream alias stream2) # hash-size 12** **(config gigastream alias stream2) # hash-bucket-id 1..3 port 1/3/q4..q6** **(config gigastream alias stream2) # comment “controlled gigastream”** **(config gigastream alias stream2) # exit**

Step	Description	Command
		(config) #
3.	Display the configuration for this example.	**(config) # show gigastream**

Advanced Hashing

Both regular GigaStream and controlled GigaStream use advanced hashing, which lets you select the criteria on which the hash is based, such as source and destination IP address, source and destination MAC address, source and destination port, and

The following table shows some different **advanced-hash** examples for regular GigaStream. Note that the **advanced-hash** method usually combines multiple criteria.

Command	Description
(config) # gigastream advanced-hash slot 3/6 fields ipdst ipsrc	Sets an **advanced-hash** method for slot 6 in box ID 3 that distributes traffic based on matching IPv4 source and destination addresses.
(config) # gigastream advanced-hash slot 7/2 default	Sets the **advanced-hash** for slot 2 in box ID 7 to the default criteria.
(config) # gigastream advanced-hash slot 1 fields macdst macsrc	Sets an **advanced-hash** method on a GigaVUE TA Series node (which only has slot 1) that distributes traffic based on matching source and destination MAC addresses.

Weighted GigaStream

Refer to the *"Weighted GigaStream"* section in the *GigaVUE Fabric Management Guide* for details about Weighted GigaStream.

Configure VLAN Manipulation

To configure a VLAN manipulation in regular by-rule maps and shared collector maps, refer to the following example:

Regular by-rule map

Map level configuration

Step	Description	Command
1.	Configure port 1/1/x1 with the type network.	**(config) # port 1/1/x1 type network**
2.	Configure port 1/1/x11 with the type tool.	**(config) # port 1/1/x11 type tool**
3.	Configure map with the alias of m2.	**(config) # map alias m2**
4.	Configure the source port for map m2.	**(config map alias m2) # from 1/1/x1**
5.	Configure the destination port for map m2.	**(config map alias m2) # to 1/1/x11**
6.	Configure the VLAN Operations at map-level: • add • modify • strip **NOTE:** For Strip, there is no need for a VLAN ID, and TPID is not required.	**(config map alias m2) # vlan-op add 100 tpid 0x9100** **(or)** **(config map alias m2) # vlan-op modify 2000 tpid 0x9100** **(or)** **(config map alias m2) # vlan-op strip**
7.	Configures rule 1 which qualifies packet with VLAN ID 200.	**(config map alias m2) # rule add pass vlan 200**
8.	Exit map configuration.	**(config map alias m2) # exit**

Rule level configuration

Step	Description	Command
1.	Configures port 1/1/x1 with the type network.	**(config) # port 1/1/x1 type network**
2.	Configures port 1/1/x11 with the type tool.	**(config) # port 1/1/x11 type tool**
3.	Configures map with the alias m2.	**(config) # map alias m2**
4.	Configure the source port for map m2.	**(config map alias m2) # from 1/1/x1**
5.	Configure the destination port for map m2.	**(config map alias m2) # to 1/1/x11**

Step	Description	Command
6.	Configure the VLAN Operations at rule-level: • add	**(config map alias m2) # rule add pass ipsrc 1.1.1.1 /32 vlan-op add 300 tpid 0x88a8**
	• modify	**(config map alias m2) # rule add pass ipsrc 2.1.1.1 /32 vlan-op modify 1000 tpid 0x8100**
	• strip	**(config map alias m2) # rule add pass ipsrc 3.1.1.1 /32 vlan-op strip**
7.	Exit map configuration.	**(config map alias m2) # exit**

Shared collector map

Map level configuration

Step	Description	Command
1.	Configures port 1/1/x1 with the type network.	**(config) # port 1/1/x1 type network**
2.	Configures port 1/1/x12 with the type tool.	**(config) # port 1/1/x12 type tool**
3.	Configures map-scollector with the alias of ms1.	**(config) # map-scollector alias ms1**
4.	configure the source port for map ms1.	**(config map alias m2) # from 1/1/x1**
5.	configure the collector port for map ms1.	**(config map alias m2) # collector 1/1/x12**
6.	Configure the VLAN Operations at map level: • add	**(config map alias m2) # vlan-op add 100 tpid 0x9100**
	• modify	**(config map alias m2) # vlan-op modify 200 tpid 0x8100**
	• strip	**vlan-op strip**
7.	Exit map-scollector configuration.	**(config map-scollector alias ms1) # exit**

Related Topics:

- To configure VLAN manipulation using GigaVUE-FM, refer to the *"VLAN Manipulation"* chapter in the *GigaVUE Fabric Management Guide*.

Configure Ingress and Egress VLAN

You can add VLAN tags to ingress packets on a per-port basis. You manually associate VLAN IDs with specific ports of type network or inline-network.

Use VLAN tags to identify, differentiate, or track incoming sources of traffic. When the traffic reaches the tools or the maps, you can filter on the VLAN tags for the corresponding ports you want to measure.

You can configure TPID for VLAN tagging. The default value of TPID is 0x8100 and other supported values are 0x9100 and 0x88a8.

The configuration examples for ingress and egress VLAN is described in the following sections:

- Ingress Port VLAN Tagging
- TPID for Ingress Port VLAN Tagging
- VLAN Tags in Maps
- Configure Egress Port VLAN Stripping

Related Topics

- Refer to the *"Using Ingress and Egress VLAN"* section in the *GigaVUE Fabric Management Guide* for details on using ingress and egress VLAN.
- Refer to the port in the reference section for details on the syntax of the commands for ingress and egress VLAN.

Ingress Port VLAN Tagging

The following example configures an ingress port VLAN tag:

Table 3: Configuring Ingress Port VLAN Tag

Command	Description
(config) # port 7/1/x1 ingress-vlan-tag 100	Configures a port with VLAN ID 100 on the specified network port.
(config) # port 1/1/x17 ingress-vlan-tag 123	Configures a port with VLAN ID 123 on the specified inline network port.

The following example replaces an ingress port VLAN ID with a new one:

Table 4: Modifying Ingress Port VLAN Tag

Command	Description
(config) # port 1/1/x1 ingress-vlan-tag 1004	Configures a port with VLAN ID 1004 on port ID 1/1/x1.
(config) # port 1/1/x1 ingress-vlan-tag 1005	Replaces VLAN ID 1004 with 1005 on port ID 1/1/x1.

The following example deletes an ingress port VLAN tag:

Table 5: Deleting Ingress Port VLAN Tag

Command	Description
(config) # no port 1/1/x1 ingress-vlan-tag	Deletes the VLAN tag associated with port ID 1/1/x1.

The following example displays ingress port VLAN tag configuration:

Table 6: Showing Ingress Port VLAN Tags

Command	Description
(config) # show port params port-list 7/1/x1	Shows the VLAN tag associated with a specific port.
(config) # show ingress-vlan-tag	Shows all configured VLAN tags.

TPID for Ingress Port VLAN Tagging

The following example configures a TPID for ingress port VLAN tag:

Table 7: Configuring Tag Protocol ID (TPID)

Command	Description
(config) # port 7/1/x1 ingress-vlan-tag 100	Configures a port with VLAN ID 100 on the specified network port.
(config) # port 7/1/x1 tag-protocol-id 0x88a8	Configures a port with TPID value on the specified network port.

The following example replaces TPID with a new one:

Table 8: Modifying Tag Protocol ID (TPID)

Command	Description
(config) # port 1/1/x1 ingress-vlan-tag 1004	Configures a port with VLAN ID 1004 on port ID 1/1/x1.
(config) # port 1/1/x1 tag-protocol-id 0x88a8	Configures a port with TPID value 0x88a8 on port ID 1/1/x1.
(config) # port 1/1/x1 tag-protocol-id 0x9100	Replaces TPID value 0x88a8 with 0x9100 on port ID 1/1/x1.

The following example deletes a TPID:

Table 9: Deleting Tag Protocol ID (TPID)

Command	Description
(config) # no port 1/1/x3 tag-protocol-id	Deletes the TPID value associated with port ID 1/1/x3.

The following example displays Tag Protocol ID (TPID) configuration:

Table 10: Showing Tag Protocol ID (TPID)

Command	Description
(config) # show tag-protocol-id	Shows all configured TPID values.

VLAN Tags in Maps

Ingress port VLAN tags are supported in first level maps, including the following:

- map
- map-passall
- map-scollector
- GigaSMART operation (gsop-enabled) maps

In the following example, the traffic from network port 2/1/q3 will be forwarded to tool port 2/1/q4. The traffic at tool port 2/1/q4 will have the added VLAN tag 1001. (Even though the VLAN tag is configured on the network port, it is added when the traffic exits the tool port.)

```
(config) # port 2/1/q3 type network
(config) # port 2/1/q3 ingress-vlan-tag 1001
(config) # port 2/1/q4 type tool
(config) # map alias m1
(config map alias m1) # type regular byRule
(config map alias m1) # from 2/1/q3
(config map alias m1) # to 2/1/q4
```

```
(config map alias m1) # rule add pass ipver 4
(config map alias m1) # exit
(config) #
```

NOTE: On GigaVUE-TA25, ingress VLAN tagging is not supported on network ports associated with GSOP maps and GSOP maps are not supported on network ports configured with ingress- vlan- tag.

Configure Egress Port VLAN Stripping

You can enable or disable outer VLAN stripping on specified egress ports. Use egress port VLAN stripping to strip an outer VLAN tag without using a GigaSMART stripping operation.

Use the **egress-vlan strip** command to enable egress port VLAN stripping. The port type must be tool or hybrid.

The following examples enable egress port VLAN stripping:

Table 11: Enabling Egress Port VLAN Stripping

Command	Description
(config) # port 1/1/c2 egress-vlan strip	Enables outer VLAN stripping on a specified egress port.
(config) # port 1/1/c3..c4 egress-vlan strip	Enables outer VLAN stripping on a range egress ports.

Once egress port VLAN stripping is enabled, it can be disabled with the **no port <port ID> egress-vlan** command.

If a port is configured for egress port VLAN stripping, the port type cannot be changed until it is disabled. To disable outer VLAN stripping on specified egress ports:

Table 12: Disabling Egress Port VLAN Stripping

Command	Description
(config) # no port 1/1/c2 egress-vlan strip	Disables outer VLAN stripping on a specified egress port.

You can view egress port VLAN stripping using the **show port params port-list bid/sid/pid** command or the **show egress-vlantag** command.

The following examples display egress port VLAN stripping configuration:

Table 13: Showing Egress Port VLAN Stripping

Command	Description
(config) # show port params port-list 1/1/c2	Shows outer VLAN stripping on a specified egress port.
(config) # show egress-vlantag	Shows the configuration of outer VLAN stripping on egress ports.

NOTE: Port level egress VLAN stripping is only supported if from/source ports are not on the same node as tool/hyrbid port with egress-vlan strip in double tag mode on GigaVUE-HC1-Plus, GigaVUE-TA25, and GigaVUE-TA25E.

Configure Inline Bypass Solutions

Security tools such as firewalls and intrusion protection systems (IPSs) are often connected inline on production networks, with traffic flowing from the network segment through the tool and back onto the production network.

Inline Bypass solutions involve bidirectional traffic between two networks, intercepted by a GigaVUE-OS node, and guided through one or more inline tools.

Inline Bypass is supported on all GigaVUE HC Series nodes: GigaVUE-HC3, and GigaVUE-HC1.

Refer to the following sections for details and examples of how to configure inline bypass solutions:

- Configuration Steps
- Configure Inline Bypass Examples

Related Topics:

- Refer to the *"Inline Bypass Solutions"* chapter in the *GigaVUE Fabric Management Guide* for more information about the inline bypass solutions.
- Refer to the *"GigaSECURE Security Delivery Platform"* section in the *GigaVUE Fabric Management Guide* for information about how the inline bypass solution supports GigaSECURE.
- Refer to the inline-tool in the reference section for details on the syntax of the inline tool CLI command.

Configuration Steps

The configuration steps in summary for an inline bypass solution are as follows:

1Configure inline network ports. (Optional for protected inline network.)

1. Configure inline network. (Optional for protected inline network.)
2. (Optional) Configure inline network group.
3. (Optional) Configure heartbeat or negative heartbeat profile.
4. Configure inline tool ports.
5. Configure inline tool.
6. (Optional) Configure inline tool group.
7. (Optional) Configure inline tool series.
8. Configure inline maps, either map passall, map (rule-based), or map shared collector.
9. Configure non-default values for parameters of the inline networks or inline tools.

The summary steps are shown in Figure 10Configuration Steps for Inline Bypass Solutions.

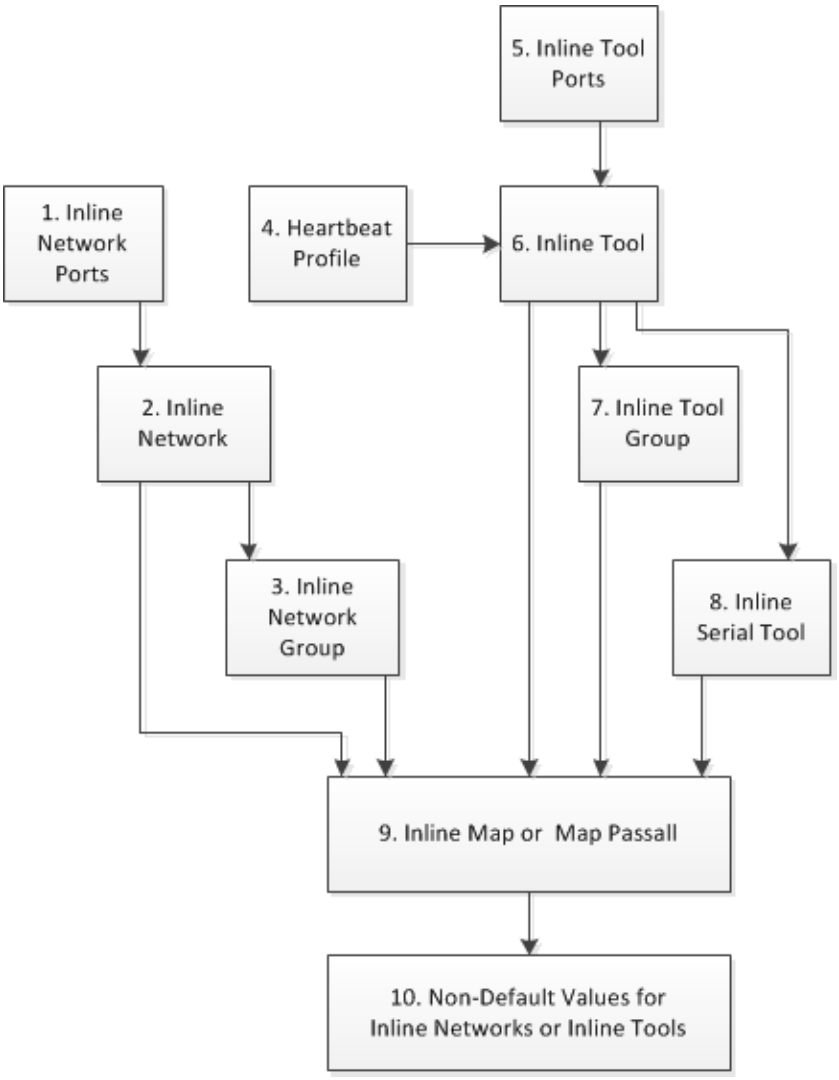


Figure 10 *Configuration Steps for Inline Bypass Solutions*

The configuration details for an inline bypass solution are as follows:

1Configure inline network ports. (Optional for protected inline network.)

The configuration begins with defining the inline network ports that will participate in the inline network. Use the **port** command with a port type of inline-network.

For an unprotected inline network, you configure the inline network ports.

For a protected inline network, the ports are created automatically when the bypass combo modules are recognized by the GigaVUE HC Series node.

10. Configure inline network. (Optional for protected inline network.)

    Next configure the inline network or inline networks using the **inline-network** command and the port pairs defined in step 1.

    For an unprotected inline network, you configure the inline network.

    For a protected inline network, the inline network is created automatically when the bypass combo modules are recognized by the GigaVUE HC Seriesnode.

    In either case, the inline network will have parameters set to default values, such as, the **traffic-path** parameter will be set to **bypass** and the **physical-bypass** parameter will be set to **enable**.

    The initial forwarding state of the unprotected inline network will be DISABLED. The initial forwarding state of the protected inline network will be PHYSICAL BYPASS.

11. (Optional) Configure inline network group.

    If the inline bypass solution involves an inline network group, first configure the participating inline networks before configuring the inline network group. Use the **inline-network-group** command and list the inline networks defined in step 10.

12. (Optional) Configure heartbeat or negative heartbeat profile.

    If any of the inline tools will be using a heartbeat profile, a default heartbeat profile is provided, so no configuration is needed except an alias. However, if any of the inline tools will be using a heartbeat profile with non-default settings, first configure the heartbeat profile using the **hb-profile** command, before configuring the inline tools that will use that profile.

    If any of the inline tools will be using a negative heartbeat profile, configure the negative heartbeat profile by providing an alias and a PCAP file using the **nhb-profile** command, before configuring the inline tools that will use that profile.

13. Configure inline tool ports.

    Next configure inline tool ports. Use the **port** command with a port type of inline-tool.

14. Configure inline tool.

    Next configure the inline tool or inline tools using the **inline-tool** command and the port pairs defined in step 13.

15. (Optional) Configure inline tool group.

If the inline bypass solution involves an inline tool group, first configure the participating inline tools, before configuring the inline tool group. Use the **inline-tool-group** command and list the inline tools defined in step 14.

16. (Optional) Configure inline tool series.

    If the inline bypass solution involves an inline tool series, first configure the participating inline tools, before configuring the inline tool series. Use the **inline-serial** command and list the inline tools defined in step 14.

17. Configure inline maps, either map passall, map (rule-based), or map shared collector.

    The next configuration step is to configure inline maps that specify how to direct the traffic from the configured inline networks and inline network groups to the configured inline tools, inline tool groups, and inline tool series. You can configure either a map passall, a map (rule-based), or a map shared collector. Use the **map**, **map-passall**, and **map-scollector** commands.

18. Configure non-default values for parameters of the inline networks or inline tools.

    Now configure non-default values for inline network parameters. For example, for an unprotected inline network, when you change the **traffic-path** parameter to **to-inline-tool**, traffic will start flowing through the inline tools from the unprotected inline network. For a protected inline network, when you change the **physical-bypass** parameter to **disable**, traffic will start flowing through the inline tools from the protected inline network.

For protected inline networks, to start the traffic flowing, perform the following steps:

1. Change the **traffic-path** parameter to **to-inline-tool**.
2. Change the **physical-bypass** parameter to **disable**.
3. Execute the following **show** commands to see if the traffic is flowing between the side A network and the side B network over a logical bypass:
   - **show port params port-list** <port ID or side A inline network port alias> and **show port params port-list** <port ID or side B inline network port alias>—The links will be *up*.
   - **show inline-network alias** <inline network alias>—The forwarding state will be FORCED BYPASS.
   - **show port stats port-list** <port ID or side A inline network port alias> and **show port stats port-list** <port ID or side B inline network port alias>—The **in** statistics for side A will match the **out** statistics for side B.

Configuration When Operationally Up

Ensure that the GigaVUE HC Series modules are in the operationally *up* state before configuring them. Configuration changes done when a module is operationally *down* are not supported.

Also, when an inline tool or inline tool group is in the operationally *down* state, do not modify the current failover action of that inline tool or inline tool group until the tool has recovered from the failover state.

Avoiding Oversubscription

In general, traffic received at inline network ports is delivered to the destination ports according to the inline maps and the out-of-band maps regardless of whether the destination ports have the capacity to absorb all the traffic or not.

> **NOTE:** When an inline network is involved in an inline map or an out-of-band map to a destination port (tool port or inline tool port), when there is temporary oversubscription, some packets arriving at the inline network port will be dropped. This can happen when the traffic path is set to bypass or monitoring.

Ensure that destination ports of maps originating from inline network ports have enough capacity to absorb the amount of traffic coming to the inline network ports.

Configure Inline Bypass Examples

The following sections provide examples of inline bypass solutions. The solutions are presented in an order from simple to complex. Refer to the following:

- Example 1: Unprotected Inline Bypass
- Example 2—Unprotected Flexible Inline, Two Collector Maps
- Example 3: Unprotected Inline Bypass with an Inline Tool Group
- Example 4: Protected Inline Bypass Using Bypass Combo Modules
- Example 5: Inline Tool Group (N+1) Redundancy
- Example 6: Inline Tool Series
- Example 7: Inline Tool Series with Local Failover Action
- Example 8: Inline Network Group (Many-to-One)
- Example 9: Inline Network Group (Many-to-Many)
- Example 10: Inline Flow Mapping® Based Solution A
- Example 11: Inline Flow Mapping® Based Solution B
- Example 12: Inline Flow Mapping® Based Solution C
- Example 13: Inline Flow Mapping® Based Solution D
- Example 14: OOB Maps Originating from Inline Network
- Example 15: OOB Maps Originating from Inline Network Group
- Example 16: Asymmetrical Hashing in Inline Tool Group
- Example 17: Maps to Individual Inline Tool Group Members
- Example 18: Gigamon Resiliency for Inline Protection

Example 1: Unprotected Inline Bypass

Example 1 is a simple, unprotected inline bypass solution. In the example, aliases are used for inline network ports (iN1 and iN2), inline tool ports (iT1 and iT2), inline network (inNet), inline tool (inTool), and inline map (inMap).

On GigaVUE-HC3, an unprotected inline bypass solution can be configured on the bypass combo module with the inline networks and inline tools on ports 1/1/x1..x16 or on ports c1..c4, or on any other module on the GigaVUE-HC3 node.

Figure 11 *Logical Bypass*

On GigaVUE-HC1, an unprotected inline bypass solution can be configured on the base module, with the inline networks and inline tools on ports 1/1/x1..x12 and 1/1/g1..g4, or on the bypass combo module on ports x1..x4.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 3/1/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 3/1/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2** **(config) # port iT2 type inline-tool**

Step	Description	Command
		(config) # port iT2 params admin enable
4.	Configure inline tool and enable it.	**(config) # inline-tool alias inTool pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool enable**
5.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inTool** **(config map-passall alias inMap) # exit**
6.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
7.	Display the configuration for this example.	**(config) # show port** **(config) # show inline-network** **(config) # show inline-tool** **(config) # show map**

Example 2: Unprotected Inline Bypass with Default Heartbeat

Example 2 adds the default heartbeat profile to the unprotected inline bypass solution on GigaVUE-HC3 in Example 1.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 3/1/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 3/1/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable**
4.	Configure default heartbeat	**(config) # hb-profile alias hb1**

Step	Description	Command
	profile.	**(config hb-profile alias hb1) # exit** **(config) #**
5.	Configure inline tool and enable it.	**(config) # inline-tool alias inTool pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool enable**
6.	Specify heartbeat profile and enable heartbeat.	**(config) # inline-tool alias inTool hb-profile hb1** **(config) # inline-tool alias inTool heart-beat**
7.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inTool** **(config map-passall alias inMap) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show hb-profile** **(config) # show inline-tool**

Example 3: Unprotected Inline Bypass with an Inline Tool Group

Example 3 adds a second inline tool to the unprotected inline bypass solution on GigaVUE-HC3 in Example 1 and creates an inline tool group consisting of two tools. It also configures a custom heartbeat profile.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 3/1/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 3/1/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2**

Step	Description	Command
		(config) # port iT2 type inline-tool **(config) # port iT2 params admin enable** **(config) # port 3/1/x5 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 3/1/x6 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable**
4.	Configure a custom heartbeat profile.	**(config) # hb-profile alias hb_custom** **(config hb-profile alias hb_custom) # custom-packet http://1.1.1.1/tftpboot/hbpackets/MyHBPacket.pcap** **(config hb-profile alias hb_custom) # packet-format custom** **(config hb-profile alias hb_custom) # exit** **(config) #**
5.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable**
6.	Specify heartbeat profile and enable heartbeat on each inline tool.	**(config) # inline-tool alias inTool1 hb-profile hb_custom** **(config) # inline-tool alias inTool2 hb-profile hb_custom** **(config) # inline-tool alias inTool1 heart-beat** **(config) # inline-tool alias inTool2 heart-beat**
7.	Configure inline tool group and enable it.	**(config) # inline-tool-group alias inToolGroup tool-list inTool1,inTool2** **(config) # inline-tool-group alias inToolGroup enable**
8.	Configure map passall, from inline network to inline tool group.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inToolGroup** **(config map-passall alias inMap) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
10.	Display the configuration for this example.	**(config) # show inline-tool-group** **(config) # show hb-profile** **(config) # show map**

Example 4: Protected Inline Bypass Using Bypass Combo Modules

Example 4 is a protected inline bypass solution using bypass combo modules on GigaVUE-HC3. It also configures heartbeat and negative heartbeat profiles.

Protected inline networks are based on the pairs of ports associated with the physical protection switches located on the bypass combo modules. Unlike the unprotected examples, you do not need to configure inline network ports because they are created automatically. On GigaVUE-HC3, the port pairs are numbered for example: 2/2/x17 and 2/2/x18, 2/2/x19 and 2/2/x20, 2/2/x21 and 2/2/x22, 2/2/x23 and 2/2/x24.

You do not need to configure inline networks because they are also created automatically on bypass combo modules. The aliases of the default inline networks are: default_inline_net_2_2_1, default_inline_net_2_2_2, default_inline_net_2_2_3, default_inline_net_2_2_4.

On GigaVUE-HC3, protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, protected inline bypass can be configured on the bypass combo module. It can also be configured on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will be 1/2/g1..g8 or 1/3/g1..g8.

NOTE: The default value of the physical-bypass attribute of protected inline networks is set to enable, which means that the fibers attached to ports net-a and net-b of the inline network are optically coupled and the traffic is exchanged between end nodes without coming to the switching fabric of the GigaVUE-OS node. As shown in Example 4, after configuring the inline tool and the map passall, the physical-bypass attribute is set to disable in order to activate the Inline Bypass solution.

Step	Description	Command
1.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 2/2/x11 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 2/2/x12 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable**
2.	Configure heartbeat profile alias.	**(config) # hb-profile alias hb2** **(config hb-profile alias hb2) # exit** **(config) #**
3.	Configure negative heartbeat profile alias and PCAP file.	**(config) # nhb-profile alias nhb1** **(config nhb-profile alias nhb1) # custom-packet**

Step	Description	Command
		http://remote/home/hnb.pcap **(config nhb-profile alias nhb1) # exit** **(config) #**
4.	Configure inline tool. Also specify the heartbeat profile, the negative heartbeat profile, enable heartbeat and negative heartbeat, and also enable inline tool.	**(config) # inline-tool alias inTool1** **(config inline-tool alias inTool1) # pair tool-a iT1 and tool-b iT2** **(config inline-tool alias inTool1) # hb-profile hb2** **(config inline-tool alias inTool1) # nhb-profile nhb1** **(config inline-tool alias inTool1) # heart-beat** **(config inline-tool alias inTool1) # negative-heart-beat** **(config inline-tool alias inTool1) # enable** **(config inline-tool alias inTool1) # exit** **(config) #**
5.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias inMap1** **(config map-passall alias inMap1) # from default_inline_net_2_2_1** **(config map-passall alias inMap1) # to inTool1** **(config map-passall alias inMap1) # exit** **(config) #**
6.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias default_inline_net_2_2_1 traffic-path to-inline-tool**
7.	Disable physical bypass on the default inline network alias.	**(config) # inline-network alias default_inline_net_2_2_1 physical-bypass disable**
8.	Display the configuration for this example.	**(config) # show port** **(config) # show inline-network** **(config) # show inline-tool** **(config) # show map** **(config) # show hb-profile** **(config) # show nhb-profile**

Example 5: Inline Tool Group (N+1) Redundancy

Example 5 is an inline bypass solution on GigaVUE-HC3 for an inline tool group with N+1 redundancy. In this example, N=2. The inline network is unprotected. Example 5 expands upon Example 3 by adding a spare to the inline tool group.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 3/1/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 3/1/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 3/1/x5 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 3/1/x6 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 3/1/x7 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 3/1/x8 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable**
5.	Configure inline tool group and parameters. Enable it and then configure failover action.	**(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) # tool-list inTool1,inTool2**

Step	Description	Command
		(config inline-tool-group alias inToolGroup) # spare-inline-tool inTool3 **(config inline-tool-group alias inToolGroup) # release-spare-if-possible** **(config inline-tool-group alias inToolGroup) # hash advanced** **(config inline-tool-group alias inToolGroup) # minimum-group-healthy-size 2** **(config inline-tool-group alias inToolGroup) # enable** **(config inline-tool-group alias inToolGroup) # failover-action tool-bypass** **(config inline-tool-group alias inToolGroup) # exit** **(config) #**
6.	Configure map passall, from inline network to inline tool group.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inToolGroup** **(config map-passall alias inMap) # exit** **(config) #**
7.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
8.	Display the configuration for this example.	**(config) # show inline-tool-group**

Example 6: Inline Tool Series

Example 6 is an inline bypass solution on GigaVUE-HC3 for an inline tool series. The inline network is unprotected. The order of the tools and inline tool groups in the tool list defines the order of the series. The map directs the traffic to the series, that is, to the first inline tool or inline tool group in the tool list. Example 6 includes two inline tools in the series and an inline tool group.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 3/1/x1 alias iN11** **(config) # port iN11 type inline-network** **(config) # port iN11 params admin enable** **(config) # port 3/1/x2 alias iN12** **(config) # port iN12 type inline-network** **(config) # port iN12 params admin enable**

Step	Description	Command
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN11 and net-b iN12**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 3/1/x5 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 3/1/x6 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 3/1/x7 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 3/1/x8 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable** **(config) # port 3/1/x9 alias iT7** **(config) # port iT7 type inline-tool** **(config) # port iT7 params admin enable** **(config) # port 3/1/x10 alias iT8** **(config) # port iT8 type inline-tool** **(config) # port iT8 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool4 pair tool-a iT7 and tool-b iT8** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable** **(config) # inline-tool alias inTool4 enable**

Step	Description	Command
5.	Configure inline tool group and parameters. Enable it and then configure failover action.	**(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) # tool-list inTool2,inTool3** **(config inline-tool-group alias inToolGroup) # enable** **(config inline-tool-group alias inToolGroup) # failover-action tool-bypass** **(config inline-tool-group alias inToolGroup) # exit** **(config) #**
6.	Configure inline tool series and enable it. Then configure failover action.	**(config) # inline-serial alias inSer** **(config inline-serial alias inSer) # inline-tool-list inTool1,inToolGroup,inTool4** **(config inline-serial alias inSer) # enable** **(config inline-serial alias inSer) # failover-action tool-bypass** **(config inline-serial alias inSer) # exit** **(config) #**
7.	Configure map passall, from inline network to inline tool series.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inSer** **(config map-passall alias inMap) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show inline-serial** **(config) # show map**

Example 7: Inline Tool Series with Local Failover Action

Example 7 is an inline bypass solution on GigaVUE-HC3 for an inline tool series. The failover action is specified for one of the inline tools (network-bypass), rather than for the series as a whole. Also the recovery mode is specified as manual.

When the individual inline tool fails, traffic is dropped at the inline network ports. When the tool recovers and is ready to be put back into service, use the **recover** command.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-	**(config) # port 3/1/x1 alias iN11** **(config) # port iN11 type inline-network**

Step	Description	Command
	network), and administratively enable inline network ports.	**(config) # port iN11 params admin enable** **(config) # port 3/1/x2 alias iN12** **(config) # port iN12 type inline-network** **(config) # port iN12 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN11 and net-b iN12**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 3/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 3/1/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 3/1/x5 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 3/1/x6 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 3/1/x7 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 3/1/x8 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable**
5.	Configure failover action and recovery mode for the second tool in the list.	**(config) # inline-tool alias inTool2 failover-action network-bypass** **(config) # inline-tool alias inTool2 recovery mode manual**
6.	Configure inline tool series,	**(config) # inline-serial alias inSer**

Step	Description	Command
	and enable it, then configure failover action, per-tool.	**(config inline-serial alias inSer) # inline-tool-list inTool1,inTool2,inTool3** **(config inline-serial alias inSer) # enable** **(config inline-serial alias inSer) # failover-action per-tool** **(config inline-serial alias inSer) # exit** **(config) #**
7.	Configure map passall, from inline network to inline tool series.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet** **(config map-passall alias inMap) # to inSer** **(config map-passall alias inMap) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show inline-tool** **(config) # show inline-serial**
10.	Display the forwarding state when the tool fails.	**(config) # show inline-network**
11.	After the inline tool recovers and is in the ready state, put the inline tool back into service.	**(config) # inline-tool alias inTool2 recover**

Example 8: Inline Network Group (Many-to-One)

Example 8 is an inline bypass solution on GigaVUE-HC3 for an inline network group. This is a many-to-one example with two inline networks and one inline tool. The inline networks are mix of protected and unprotected.

On GigaVUE-HC3, unprotected inline bypass can be configured on any module on the node. Protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, unprotected inline bypass can be configured on the base module, with the inline networks and inline tools on ports 1/1/x1..x12 and 1/1/g1..g4, or on the bypass combo module on ports x1..x4. Protected inline bypass can be configured on the bypass combo module, or on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will be 1/2/g1..g8 or 1/3/g1..g8. On the TAP module, you will need to configure inline network ports and the inline network because they are not created automatically (as they are on bypass combo modules).

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 7/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 7/2/x20 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure an inline network group consisting of a single unprotected inline network and two protected inline networks.	**(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) # network-list inNet,default_inline_net_7_2_1,default_inline_net_7_2_3** **(config inline-network-group alias inNetGroup) # exit** **(config) #**
4.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable**
5.	Configure inline tool and enable it. Also, specify that the inline tool is going to be shared by different sources. When shared is enabled (true), the inline tool can receive traffic from multiple sources (the inline networks in the inline network group).	**(config) # inline-tool alias inTool pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool enable** **(config) # inline-tool alias inTool shared true**
6.	Configure map passall, from inline network group to inline tool.	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNetGroup** **(config map-passall alias inMap) # to inTool** **(config map-passall alias inMap) # exit** **(config) #**
7.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_7_2_1 traffic-path to-inline-tool**

Step	Description	Command
		(config) # inline-network alias default_inline_net_7_2_3 traffic-path to-inline-tool
8.	Disable physical bypass on the default inline network aliases.	**(config) # inline-network alias default_inline_net_7_2_1 physical-bypass disable** **(config) # inline-network alias default_inline_net_7_2_3 physical-bypass disable**
9.	Display the configuration for this example.	**(config) # show inline-network-group** **(config) # show inline-tool** **(config) # show map**

Example 9: Inline Network Group (Many-to-Many)

Example 9 is an inline bypass solution on GigaVUE-HC3 for an inline network group. Example 9 expands upon Example 8 by adding a second inline tool. The inline networks are a mix of unprotected and protected.

In addition, user-defined VLAN tags are added in Example 9 to guide traffic from the multiple inline networks in the inline network group.

On GigaVUE-HC3, unprotected inline bypass can be configured on any module on the node. Protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, unprotected inline bypass can be configured on the base module, with the inline networks and inline tools on ports 1/1/x1..x12 and 1/1/g1..g4, or on the bypass combo module on ports x1..x4. Protected inline bypass can be configured on the bypass combo module, or on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will be 1/2/g1..g8 or 1/3/g1..g8. On the TAP module, you will need to configure inline network ports and the inline network because they are not created automatically (as they are on bypass combo modules).

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 7/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 7/2/x20 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**

Step	Description	Command
3.	Configure an inline network group consisting of a single unprotected inline network and two protected inline networks.	**(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) # network-list inNet,default_inline_net_7_2_1,default_inline_net_7_2_3** **(config inline-network-group alias inNetGroup) # exit** **(config) #**
4.	(Optional) Configure user-defined VLAN tags. **Note**: The net-a and net-b ports can have the same VLAN tag, but tags must otherwise be unique within the inline network group.	**(config) # port 7/2/x1 ingress-vlan-tag 1201** **(config) # port 7/2/x20 ingress-vlan-tag 1202** **(config) # port 7/2/x17 ingress-vlan-tag 1203** **(config) # port 7/2/x18 ingress-vlan-tag 1203**
5.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **7/2/x9 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **7/2/x10 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable**
6.	Configure inline tools and enable them. Also, specify that inline tools are going to be shared by different sources. When shared is enabled (true), the inline tools can receive traffic from multiple sources (the inline networks in the inline network group).	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **inTool1 shared true** **(config) # inline-tool alias inTool2 shared true**
7.	Configure inline tool group and enable it.	**(config) # inline-tool-group alias inToolGroup tool-list inTool1,inTool2** **(config) # inline-tool-group alias inToolGroup enable**
8.	Configure map passall, from inline network to	**(config) # map-passall alias inMap** **(config map-passall alias inMap) # from inNet**

Step	Description	Command
	inline tool group.	**(config map-passall alias inMap) # to inToolGroup** **(config map-passall alias inMap) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_7_2_1 traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_7_2_3 traffic-path to-inline-tool**
10.	Disable physical bypass on the default inline network aliases.	**(config) # inline-network alias default_inline_net_7_2_1 physical-bypass disable** **(config) # inline-network alias default_inline_net_7_2_3 physical-bypass disable**
11.	Display the configuration for this example.	**(config) # show inline-network-group** **(config) # show ingress-vlan-tag** **(config) # show inline-tool-group**

Example 10: Inline Flow Mapping® Based Solution A

Example 10 is an inline Flow Mapping based solution on GigaVUE-HC3. Example 10 has a single, unprotected inline network, a single inline tool, a rule-based map (VLAN 100) from the inline network to the inline tool, and a shared collector from the inline network to bypass. Traffic on VLAN 100 will be inspected by the inline tool while the remaining traffic will not be inspected (will be bypassed).

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 7/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 7/2/x20 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x2 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x15 alias iT2**

Step	Description	Command
		(config) # port iT2 type inline-tool **(config) # port iT2 params admin enable**
4.	Configure inline tool and enable it.	**(config) # inline-tool alias inTool pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool enable**
5.	Enable default heartbeat.	**(config) # inline-tool alias inTool heart-beat**
6.	Configure rule-based map, from inline network to inline tool.	**(config) # map alias inMap1** **(config map alias inMap1) # type inline byRule** **(config map alias inMap1) # from inNet** **config map alias inMap1) # to inTool** **(config map alias inMap1) # rule add pass vlan 100** **(config map alias inMap1) # exit** **(config) #**
7.	Add a shared collector for any unmatched data and send it to bypass.	**(config) # map-scollector alias scoll** **(config map-scollector alias scoll) # from inNet** **(config map-scollector alias scoll) # collector bypass** **(config map-scollector alias scoll) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show inline-network** **(config) # show inline-tool** **(config) # show map**

Example 11: Inline Flow Mapping® Based Solution B

Example 11 is an inline Flow Mapping based solution on GigaVUE-HC3. Example 11 has a single, unprotected inline network, a single inline tool, a rule-based map (VLAN 100) from the inline network to bypass, and a shared collector from the inline network to the inline tool. Traffic on VLAN 100 will not be inspected by the inline tool, while the remaining traffic will be inspected by the inline tool (through the bypass).

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable	**(config) # port 7/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable**

Step	Description	Command
	inline network ports.	**(config) # port 7/2/x20 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x2 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x15 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable**
4.	Configure inline tool and enable it.	**(config) # inline-tool alias inTool pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool enable**
5.	Enable default heartbeat.	**(config) # inline-tool alias inTool heart-beat**
6.	Configure rule-based map, from inline network to bypass.	**(config) # map alias inMap2** **(config map alias inMap2) # type inline byRule** **(config map alias inMap2) # from inNet** **(config map alias inMap2) # to bypass** **(config map alias inMap2) # rule add pass vlan 100** **(config map alias inMap2) # exit** **(config) #**
7.	Add a shared collector, from inline network to inline tool.	**(config) # map-scollector alias scoll2** **(config map-scollector alias scoll2) # from inNet** **(config map-scollector alias scoll2) # collector inTool** **(config map-scollector alias scoll2) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show map**

Example 12: Inline Flow Mapping® Based Solution C

Example 12 is an inline Flow Mapping based solution on GigaVUE-HC3. Example 12 has a single, unprotected inline network, two individual inline tools, a rule-based map (portdst 22) from the inline network to bypass, a rule-based map (portdst 80) from the inline network to

the first inline tool, and a shared collector from the inline network to the second inline tool. Traffic that does not match the map rules will be sent to the shared collector, ensuring that all traffic is exchanged between side A and side B of the network.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 7/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 7/2/x20 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x2 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x15 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 7/2/x3 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 7/2/x4 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable**
5.	Enable default heartbeats.	**(config) # inline-tool alias inTool1 heart-beat** **(config) # inline-tool alias inTool2 heart-beat**
6.	Configure rule-based map, from inline network to bypass.	**(config) # map alias inMap3** **(config map alias inMap3) # type inline byRule** **(config map alias inMap3) # from inNet** **(config map alias inMap3) # to bypass** **(config map alias inMap3) # rule add pass portdst 22** **(config map alias inMap3) # exit**

Step	Description	Command
		(config) #
7.	Configure rule-based map, from inline network to first inline tool.	**(config) # map alias inMap4** **(config map alias inMap4) # type inline byRule** **(config map alias inMap4) # from inNet** **(config map alias inMap4) # to inTool1** **(config map alias inMap4) # rule add pass portdst 80** **(config map alias inMap4) # exit** **(config) #**
8.	Add a shared collector, from inline network to second inline tool.	**(config) # map-scollector alias scoll3** **(config map-scollector alias scoll3) # from inNet** **(config map-scollector alias scoll3) # collector inTool2** **(config map-scollector alias scoll3) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
10.	Display the configuration for this example.	**(config) # show inline-tool** **(config) # show map**

Example 13: Inline Flow Mapping® Based Solution D

Example 13 is an inline Flow Mapping based solution on GigaVUE-HC3. Example 13 has a variety of constructs: an inline network group made up of two protected inline networks, an inline tool group, an inline tool series, an individual inline tool, a rule-based map (VLAN 100) from the inline network group to the inline tool group, a rule-based map (portdst 80) from the inline network group to the inline tool series, a rule-based map (ipsrc 10.123.12.57) from the inline network group to the individual inline tool, and a shared collector from the inline network group to bypass.

Since Example 13 uses protected inline networks on GigaVUE-HC3, they do not need to be configured as described in Example 4: Protected Inline Bypass Using Bypass Combo Modules, so the configuration begins with the inline network group.

On GigaVUE-HC3, unprotected inline bypass can be configured on any module on the node. Protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, unprotected inline bypass can be configured on the base module, with the inline networks and inline tools on ports 1/1/x1..x12 and 1/1/g1..g4, or on the bypass combo module on ports x1..x4. Protected inline bypass can be configured on the bypass combo module, or on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will

be 1/2/g1..g8 or 1/3/g1..g8. On the TAP module, you will need to configure inline network ports and the inline network because they are not created automatically (as they are on bypass combo modules).

Step	Description	Command
1.	Configure an inline network group consisting of two protected inline networks.	**(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) # network-list default_inline_net_7_2_1,default_inline_net_7_2_3** **(config inline-network-group alias inNetGroup) # exit** **(config) #**
2.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 7/2/x2 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 7/2/x15 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 7/2/x3 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 7/2/x4 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 7/2/x7 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 7/2/x8 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable** **(config) # port 7/2/x13 alias iT7** **(config) # port iT7 type inline-tool** **(config) # port iT7 params admin enable** **(config) # port 7/2/x14 alias iT8** **(config) # port iT8 type inline-tool** **(config) # port iT8 params admin enable** **(config) # port 7/2/x15 alias iT9** **(config) # port iT9 type inline-tool** **(config) # port iT9 params admin enable** **(config) # port 7/2/x16 alias iT10** **(config) # port iT10 type inline-tool** **(config) # port iT10 params admin enable**

Step	Description	Command
3.	Configure inline tools as follows: • inTool1 and inTool2 will be used in the inline tool group, inToolGroup • inTool3 will be the individual inline tool used in Map3 • inTool4 and inTool5 will be used in the inline tool series, inSer Also, enable inline tools. Specify that inline tools are going to be shared by different sources. When shared is enabled (true), the inline tools can receive traffic from multiple sources (the inline networks in the inline network group).	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool4 pair tool-a iT7 and tool-b iT8** **(config) # inline-tool alias inTool5 pair tool-a iT9 and tool-b iT10** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable** **(config) # inline-tool alias inTool4 enable** **(config) # inline-tool alias inTool5 enable** **inTool1 shared true** **(config) # inline-tool alias inTool2 shared true** **(config) # inline-tool alias inTool3 shared true** **(config) # inline-tool alias inTool4 shared true** **(config) # inline-tool alias inTool5 shared true**
4.	Enable default heartbeats.	**(config) # inline-tool alias inTool1 heart-beat** **(config) # inline-tool alias inTool2 heart-beat** **(config) # inline-tool alias inTool3 heart-beat** **(config) # inline-tool alias inTool4 heart-beat** **(config) # inline-tool alias inTool5 heart-beat**
5.	Configure an inline tool group and enable it.	**(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) # tool-list inTool1,inTool2** **(config inline-tool-group alias inToolGroup) # enable** **(config inline-tool-group alias inToolGroup) # exit** **(config) #**
6.	Configure an inline tool series and enable it.	**(config) # inline-serial alias inSer** **(config inline-serial alias inSer) # inline-tool-list inTool4,inTool5** **(config inline-serial alias inSer) # enable** **(config inline-serial alias inSer) # exit** **(config) #**
7.	Configure rule-based map, from the inline network	**(config) # map alias inMap1** **(config map alias inMap1) # type inline byRule**

Step	Description	Command
	group to the inline tool group.	**(config map alias inMap1) # from inNetGroup** **(config map alias inMap1) # to inToolGroup** **(config map alias inMap1) # rule add pass vlan 100** **(config map alias inMap1) # exit** **(config) #**
8.	Configure rule-based map, from the inline network group to the inline tool series.	**(config) # map alias inMap2** **(config map alias inMap2) # type inline byRule** **(config map alias inMap2) # from inNetGroup** **(config map alias inMap2) # to inSer** **(config map alias inMap2) # rule add pass portdst 80** **(config map alias inMap2) # exit** **(config) #**
9.	Configure rule-based map, from the inline network group to the individual inline tool.	**(config) # map alias inMap3** **(config map alias inMap3) # type inline byRule** **(config map alias inMap3) # from inNetGroup** **(config map alias inMap3) # to inTool3** **(config map alias inMap3) # rule add pass ipsrc 10.123.12.57 255.255.255.248** **(config map alias inMap3) # exit** **(config) #**
10.	Add a shared collector from the inline network group to bypass.	**(config) # map-scollector alias scoll** **(config map-scollector alias scoll) # from inNetGroup** **(config map-scollector alias scoll) # collector bypass** **(config map-scollector alias scoll) # exit** **(config) #**
11.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_7_2_1 traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_7_2_3 traffic-path to-inline-tool**
12.	Disable physical bypass on the default inline network aliases.	**(config) # inline-network alias default_inline_net_7_2_1 physical-bypass disable** **(config) # inline-network alias default_inline_net_7_2_3 physical-bypass disable**
13.	Display the configuration for this example.	**(config) # show inline-network** **(config) # show inline-network-group** **(config) # show inline-tool**

Step	Description	Command
		(config) # show inline-serial **(config) # show inline-tool-group** **(config) # show map**

Example 14: OOB Maps Originating from Inline Network

Example 14 combines out-of-band (OOB) maps with a map passall originating from an inline network on GigaVUE-HC3. In Example 14, the map passall sends all traffic to the inline tool. The OOB rule-based map sends traffic to an OOB tool.

When the source port of an OOB map is associated with an inline network, multiple source ports are supported in the port list (the **from** argument of the **map** command).

A protected inline network (which uses bypass combo modules) is included in Example 14. You do not need to configure inline network ports because they are created automatically. The port pairs in Example 14 are 1/1/x21 and 1/1/x22. You do not need to configure an inline network because it is also created automatically. The alias of the default inline network in Example 14 is default_inline_net_1_1_3.

On GigaVUE-HC3, protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, protected inline bypass can be configured on the bypass combo module, or on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will be 1/2/g1..g8 or 1/3/g1..g8. On the TAP module, you will need to configure inline network ports and the inline network because they are not created automatically (as they are on bypass combo modules).

Step	Description	Command
1.	Configure a regular tool port of port type (tool) and administratively enable it. This is the OOB tool.	**(config) # port 1/1/x12 type tool** **(config) # port 1/1/x12 params admin enable**
2.	Configure two inline tool ports of port type (inline-tool) and administratively enable them.	**(config) # port 1/2/x23 type inline-tool** **(config) # port 1/2/x23 params admin enable** **(config) # port 1/2/x24 type inline-tool** **(config) # port 1/2/x24 params admin enable**
3.	Configure inline tool and enable it.	**(config) # inline-tool alias inTool1 pair tool-a 1/2/x23 and tool-b 1/2/x24**

Step	Description	Command
		(config) # inline-tool alias inTool1 enable
4.	Configure a map passall, from the inline network to the inline tool. This sends all the traffic to the inline tool.	**(config) # map-passall alias inline_map1** **(config map-passall alias inline_map1) # from default_ inline_net_1_1_3** **(config map-passall alias inline_map1) # to inTool1** **(config map-passall alias inline_map1) # exit** **(config) #**
5.	Configure the OOB rule-based map, with both inline network ports in the **from** argument, and the OOB tool in the **to** argument.	**(config) # map alias OoB_map** **(config map alias OoB_map) # type regular byRule** **(config map alias OoB_map) # rule add pass ipver 4** **(config map alias OoB_map) # to 1/1/x12** **(config map alias OoB_map) # from 1/1/x21..x22** **(config map alias OoB_map) # exit** **(config) #**
6.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias default_inline_net_1_1_3 traffic-path to-inline-tool**
7.	Disable physical bypass on the default inline network alias.	**(config) # inline-network alias default_inline_net_1_1_3 physical-bypass disable**
8.	Display the configuration and statistics for this example.	**(config) # show inline-network** **(config) # show inline-tool** **(config) # show map** **(config) # show port stats**

Example 15: OOB Maps Originating from Inline Network Group

Example 15 expands on Example 14 by combining out-of-band (OOB) maps with a map passall originating from an inline network group on GigaVUE-HC3.

When the source port of an OOB map is associated with an inline network group, only one port is supported in the port list. In this case, multiple OOB maps are needed because each OOB map only accepts one inline network port as the input (the **from** argument of the **map** command).

A protected inline network (which uses bypass combo modules) is included in Example 15. You do not need to configure inline network ports or the inline networks because they are created automatically. The port pairs in Example 15 are 1/1/x17 and 1/1/x18, as well as 1/1/x19 and 1/1/x20. The aliases of the default inline networks in Example 15 are default_inline_net_1_1_1 and default_inline_net_1_1_2.

In Example 15, two OOB maps send traffic from each inline network port (associated with default_inline_net_1_1_1) to the OOB tool. Two more maps would be needed to send traffic from each inline network port (associated with default_inline_net_1_1_2) to the OOB tool, but this is not included in Example 15.

On GigaVUE-HC3, protected inline bypass can be configured on the bypass combo module on ports c1..c4.

On GigaVUE-HC1, protected inline bypass can be configured on the bypass combo module, or on the TAP-HC1-G10040 module placed in either bay 2 or bay 3, so the ports will be 1/2/g1..g8 or 1/3/g1..g8. On the TAP module, you will need to configure inline network ports and the inline network because they are not created automatically (as they are on bypass combo modules).

Step	Description	Command
1.	Configure an inline network group consisting of two protected inline networks.	**(config) # inline-network-group alias inNetGroup** **(config inline-network-group alias inNetGroup) # network-list default_inline_net_1_1_1,default_inline_net_1_1_2** **(config inline-network-group alias inNetGroup) # exit** **(config) #**
2.	Configure a regular tool port of port type (tool) and administratively enable it. This is the OOB tool.	**(config) # port 1/1/x12 type tool** **(config) # port 1/1/x12 params admin enable**
3.	Configure two inline tool ports of port type (inline-tool) and administratively enable them.	**(config) # port 1/2/x23 type inline-tool** **(config) # port 1/2/x23 params admin enable** **(config) # port 1/2/x24 type inline-tool** **(config) # port 1/2/x24 params admin enable**
4.	Configure inline tool and enable it. Also, specify that the inline tool is going to be shared by different sources. When shared is enabled (true), the inline tool can receive traffic from multiple sources (the inline networks in the inline network group).	**(config) # inline-tool alias inTool1 pair tool-a 1/2/x23 and tool-b 1/2/x24** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool1 shared true**
5.	Configure a map passall, from the inline network group to the inline tool. This sends all the traffic to the	**(config) # map-passall alias inline_map1** **(config map-passall alias inline_map1) # from inNetGroup** **(config map-passall alias inline_map1) # to inTool1**

Step	Description	Command
	inline tool.	**(config map-passall alias inline_map1) # exit** **(config) #**
6.	Configure the first rule-based map. This is an OOB map from one inline network port (associated with default_inline_net_1_1_1) to the OOB tool.	**(config) # map alias OoB_map1** **(config map alias OoB_map1) # type regular byRule** **(config map alias OoB_map1) # rule add pass ipver 4** **(config map alias OoB_map1) # to 1/1/x12** **(config map alias OoB_map1) # from 1/1/x17** **(config map alias OoB_map1) # exit** **(config) #**
7.	Configure a second rule-based map. This is an OOB map from the other inline network port (associated with default_inline_net_1_1_1) to the OOB tool.	**(config) # map alias OoB_map2** **(config map alias OoB_map2) # type regular byRule** **(config map alias OoB_map2) # rule add pass ipver 4** **(config map alias OoB_map2) # to 1/1/x12** **(config map alias OoB_map2) # from 1/1/x18** **(config map alias OoB_map2) # exit** **(config) #**
8.	Configure a third rule-based map. This is an OOB map from a single inline tool port to the OOB tool.	**(config) # map alias OoB_map3** **(config map alias OoB_map3) # type inline byRule** **(config map alias OoB_map3) # rule add pass ipver 4** **(config map alias OoB_map3) # to 1/1/x12** **(config map alias OoB_map3) # from 1/2/x23** **(config map alias OoB_map3) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias default_inline_net_1_1_1 traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_1_1_2 traffic-path to-inline-tool**
10.	Disable physical bypass on the default inline network aliases.	**(config) # inline-network alias default_inline_net_1_1_1 physical-bypass disable** **(config) # inline-network alias default_inline_net_1_1_2 physical-bypass disable**
11.	Display the configuration and statistics for this example.	**(config) # show inline-network** **(config) # show inline-network-group** **(config) # show inline-tool** **(config) # show map**

Example 16: Asymmetrical Hashing in Inline Tool Group

Example 16 is an inline bypass solution on GigaVUE-HC3 for an inline tool group with four tools. The inline tool group uses asymmetrical hashing (unlike Example 5: Inline Tool Group (N+1) Redundancy which uses symmetrical hashing). The hashing is based on the source IP address for side A and the destination IP address for side B.

A rule-based map (vlan 200) is configured from the inline network to the inline tool group. Traffic that matches the map rule and has the same source IP on side A and destination IP on side B will be sent to the same inline tool in the inline tool group.

A shared collector is configured from the inline network to bypass. Traffic that does not match the map rule will be sent to the shared collector and bypassed.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 1/2/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/2/x15 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 1/2/x16 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 1/2/x19 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 1/2/x20 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 1/2/x21 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 1/2/x22 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable**

Step	Description	Command
		(config) # port 1/2/x23 alias iT7(**config) # port iT7 type inline-tool** **(config) # port iT7 params admin enable** **(config) # port 1/2/x24 alias iT8** **(config) # port iT8 type inline-tool** **(config) # port iT8 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool4 pair tool-a iT7 and tool-b iT8** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable** **(config) # inline-tool alias inTool4 enable**
5.	Configure inline tool group and parameters. Enable it and then configure failover action.	**(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) # tool-list inTool1,inTool2,inTool3,inTool4** **(config inline-tool-group alias inToolGroup) # hash a-srcip-b-dstip** **(config inline-tool-group alias inToolGroup) # minimum-group-healthy-size 4** **(config inline-tool-group alias inToolGroup) # enable** **(config inline-tool-group alias inToolGroup) # failover-action tool-bypass** **(config inline-tool-group alias inToolGroup) # exit** **(config) #**
6.	Configure rule-based map, from inline network to inline tool group.	**(config) # map alias inNet-to-ITG** **(config map alias inNet-to-ITG) # type inline byRule** **(config map alias inNet-to-ITG) # from inNet** **(config map alias inNet-to-ITG) # to inToolGroup** **(config map alias inNet-to-ITG) # rule add pass vlan 200** **(config map alias inNet-to-ITG) # exit** **(config) #**
7.	Add a shared collector for any unmatched data and	**(config) # map-scollector alias inNet-to-bypass** **(config map-scollector alias inNet-to-bypass) # from**

Step	Description	Command
	send it to bypass.	**inNet** **(config map-scollector alias inNet-to-bypass) # collector bypass** **(config map-scollector alias inNet-to-bypass) # exit** **(config) #**
8.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
9.	Display the configuration for this example.	**(config) # show inline-tool-group** **(config) # show map**

Example 17: Maps to Individual Inline Tool Group Members

Example 17 is an inline bypass solution on GigaVUE-HC3 for an inline tool group with four tools. It is similar to Example 16: Asymmetrical Hashing in Inline Tool Group, but has four rule-based inline maps, one to each individual member of the inline tool group. In Example 17, asymmetrical hashing is used, but the hashing could also be symmetrical. The hashing only applies to the traffic sent to the shared collector.

Example 17 is different from Example 5: Inline Tool Group (N+1) Redundancy. In Example 5, all the traffic was sent to the inline tool group as a whole, using a map passall. Hashing distributed the traffic across the group.

With the multiple rule-based maps in Example 17, specific traffic is sent to specific tools in the inline tool group according to the rules. Each of the four inline maps directs traffic from one source IP address to a specific inline tool in the group.

A shared collector is configured from the inline network to the inline tool group. Traffic that does not match any of the map rules is sent to the shared collector and will be distributed according to the hashing value specified for the group.

Step	Description	Command
1.	Configure inline network aliases, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 1/2/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias inNet pair net-a iN1 and net-b iN2**

Step	Description	Command
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/2/x15 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 1/2/x16 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable** **(config) # port 1/2/x19 alias iT3** **(config) # port iT3 type inline-tool** **(config) # port iT3 params admin enable** **(config) # port 1/2/x20 alias iT4** **(config) # port iT4 type inline-tool** **(config) # port iT4 params admin enable** **(config) # port 1/2/x21 alias iT5** **(config) # port iT5 type inline-tool** **(config) # port iT5 params admin enable** **(config) # port 1/2/x22 alias iT6** **(config) # port iT6 type inline-tool** **(config) # port iT6 params admin enable** **(config) # port 1/2/x23 alias iT7** **(config) # port iT7 type inline-tool** **(config) # port iT7 params admin enable** **(config) # port 1/2/x24 alias iT8** **(config) # port iT8 type inline-tool** **(config) # port iT8 params admin enable**
4.	Configure inline tools and enable them.	**(config) # inline-tool alias inTool1 pair tool-a iT1 and tool-b iT2** **(config) # inline-tool alias inTool2 pair tool-a iT3 and tool-b iT4** **(config) # inline-tool alias inTool3 pair tool-a iT5 and tool-b iT6** **(config) # inline-tool alias inTool4 pair tool-a iT7 and tool-b iT8** **(config) # inline-tool alias inTool1 enable** **(config) # inline-tool alias inTool2 enable** **(config) # inline-tool alias inTool3 enable** **(config) # inline-tool alias inTool4 enable**
5.	Configure inline tool group and parameters. Enable it and then configure failover	**(config) # inline-tool-group alias inToolGroup** **(config inline-tool-group alias inToolGroup) # tool-list inTool1,inTool2,inTool3,inTool4**

Step	Description	Command
	action.	**(config inline-tool-group alias inToolGroup) # hash a-srcip-b-dstip** **(config inline-tool-group alias inToolGroup) # minimum-group-healthy-size 4** **(config inline-tool-group alias inToolGroup) # enable** **(config inline-tool-group alias inToolGroup) # failover-action network-bypass** **(config inline-tool-group alias inToolGroup) # exit** **(config) #**
6.	Configure rule-based map, from inline network to first tool in inline tool group, from the same source, inNet.	**(config) # map alias inNet-to-inTool1** **(config map alias inNet-to-inTool1) # type inline byRule** **(config map alias inNet-to-inTool1) # from inNet** **(config map alias inNet-to-inTool1) # to inTool1** **(config map alias inNet-to-inTool1) # rule add pass ipsrc 10.10.10.101 /32** **(config map alias inNet-to-inTool1) # exit** **(config) #**
7.	Configure rule-based map, from inline network to second tool in inline tool group, from the same source, inNet.	**(config) # map alias inNet-to-inTool2** **(config map alias inNet-to-inTool2) # type inline byRule** **(config map alias inNet-to-inTool2) # from inNet** **(config map alias inNet-to-inTool2) # to inTool2** **(config map alias inNet-to-inTool2) # rule add pass ipsrc 20.10.20.102 /32** **(config map alias inNet-to-inTool2) # exit** **(config) #**
8.	Configure rule-based map, from inline network to third tool in inline tool group, from the same source, inNet.	**(config) # map alias inNet-to-inTool3** **(config map alias inNet-to-inTool3) # type inline byRule** **(config map alias inNet-to-inTool3) # from inNet** **(config map alias inNet-to-inTool3) # to inTool3** **(config map alias inNet-to-inTool3) # rule add pass ipsrc 31.11.31.103 /32** **(config map alias inNet-to-inTool3) # exit** **(config) #**
9.	Configure rule-based map, from inline network to fourth tool in inline tool group, from the same source, inNet.	**(config) # map alias inNet-to-inTool4** **(config map alias inNet-to-inTool4) # type inline byRule** **(config map alias inNet-to-inTool4) # from inNet** **(config map alias inNet-to-inTool4) # to inTool4** **(config map alias inNet-to-inTool4) # rule add pass ipsrc 41.11.41.104 /32**

Step	Description	Command
		(config map alias inNet-to-inTool4) # exit **(config) #**
10.	Add a shared collector for any unmatched data and send it to the inline tool group. Again, the source is the same, inNet.	**(config) # map-scollector alias inNet-to-ITG** **(config map-scollector alias inNet-to-ITG) # from inNet** **(config map-scollector alias inNet-to-ITG) # collector inToolGroup** **(config map-scollector alias inNet-to-ITG) # exit** **(config) #**
11.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inNet traffic-path to-inline-tool**
12.	Display the configuration for this example.	**(config) # show inline-tool-group** **(config) # show map**

Example 18: Gigamon Resiliency for Inline Protection

You can configure Gigamon Resiliency for inline protection on H Series nodes (GigaVUE-HC1, and GigaVUE-HC3). Example 18 is an inline bypass solution for GRIP using TAP-HC1-G10040 modules on GigaVUE-HC1 with copper ports. The same instructions apply to GigaVUE-HC3.

First, configure the GigaVUE-HC1 with the primary role, then configure the GigaVUE-HC1 with the secondary role. The configuration is the same (is synchronized) on both nodes, except for step 3, in which the protection role (primary or secondary) is specified.

Note that in this example, link fail propagation (LFP) is disabled to reduce inline network recovery time after failover. When a primary to secondary failover occurs and LFP is enabled for copper inline bypass links, network service recovery may take several seconds because of Ethernet link renegotiation. Optical links failover faster and typically recover service much faster. For inline networks where only one path is available, this is a consideration. When GRIP is deployed with high availability networks where a second path is present, it is a best practice to leave LFP enabled.

Configuring Primary Role GigaVUE-HC1

Step	Description	Command
1.	Configure ports on the TAP-HC1-G10040 module as passive (in passive mode, relays are closed). Also configure ports, port type (inline-network).	**(config) # port 1/3/g1..g8 params taptx passive** **(config) # port 1/3/g1..g8 type inline-network**

Step	Description	Command
2.	Configure stack port (for signaling port/link) and enable it.	**(config) # port 1/1/x1 type stack** **(config) # port 1/1/x1 params admin enable**
3.	Create the redundancy profile by giving it a name and configuring parameters for the redundancy profile such as the signaling port and protection role (primary).	**(config) # redundancy-profile alias RP_001 (config redundancy-profile alias RP_001) # signaling-port 1/1/x1** **(config redundancy-profile alias RP_001) # protection-role primary** **(config redundancy-profile alias RP_001) # exit** **(config) #**
4.	Configure inline network.	**(config) # inline-network alias IN_001 pair net-a 1/3/g1 and net-b 1/3/g2**
5.	Associate the redundancy profile to the inline network. Also disable link fail propagation on the inline network.	**(config) # inline-network alias IN_001 redundancy-profile RP_001** **(config) # no inline-network alias IN_001 lfp enable**
6.	Configure inline tool ports, port type (inline-tool), and administratively enable them.	**(config) # port 1/1/x11 type inline-tool** **(config) # port 1/1/x11 params admin enable** **(config) # port 1/1/x12 type inline-tool** **(config) # port 1/1/x12 params admin enable**
7.	Configure inline tool and failover action. Then enable inline tool.	**(config) # inline-tool alias IT_001 pair tool-a 1/1/x11 and tool-b 1/1/x12** **(config) # inline-tool alias IT_001 failover-action network-bypass** **(config) # inline-tool alias IT_001 enable**
8.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias INtoIT** **(config map-passall alias INtoIT) # from IN_001** **(config map-passall alias INtoIT) # to IT_001** **(config map-passall alias INtoIT) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias IN_001 traffic-path to-inline-tool**

Configure Secondary Role GigaVUE-HC1

Step	Description	Command
1.	Configure ports on the TAP-HC1-G10040 module as	**(config) # port 1/3/g1..g8 params taptx passive** **(config) # port 1/3/g1..g8 type inline-network**

Step	Description	Command
	passive (in passive mode, relays are closed). Also configure ports, port type (inline-network).	
2.	Configure stack port (for signaling port/link) and enable it.	**(config) # port 1/1/x1 type stack** **(config) # port 1/1/x1 params admin enable**
3.	Configure parameters for the redundancy profile such as the signaling port and protection role (secondary).	**(config) # redundancy-profile alias RP_001 (config redundancy-profile alias RP_001) # signaling-port 1/1/x1** **(config redundancy-profile alias RP_001) # protection-role secondary** **(config redundancy-profile alias RP_001) # exit** **(config) #**
4.	Configure inline network.	**(config) # inline-network alias IN_001 pair net-a 1/3/g1 and net-b 1/3/g2**
5.	Associate the redundancy profile to the inline network. Also disable link fail propagation on the inline network.	**(config) # inline-network alias IN_001 redundancy-profile RP_001** **(config) # no inline-network alias IN_001 lfp enable**
6.	Configure inline tool ports, port type (inline-tool), and administratively enable them.	**(config) # port 1/1/x11 type inline-tool** **(config) # port 1/1/x11 params admin enable** **(config) # port 1/1/x12 type inline-tool** **(config) # port 1/1/x12 params admin enable**
7.	Configure inline tool and failover action. Then enable inline tool.	**(config) # inline-tool alias IT_001 pair tool-a 1/1/x11 and tool-b 1/1/x12** **(config) # inline-tool alias IT_001 failover-action network-bypass** **(config) # inline-tool alias IT_001 enable**
8.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias INtoIT** **(config map-passall alias INtoIT) # from IN_001** **(config map-passall alias INtoIT) # to IT_001** **(config map-passall alias INtoIT) # exit** **(config) #**
9.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias IN_001 traffic-path to-inline-tool**

Configure Inline Bypass Solution on GigaVUE-OS TAP Modules

Network ports on the copper TAP modules, TAP-HC1-G10040 on the GigaVUE-HC1 can be configured through software to be inline network ports. This allows the GigaVUE-OS TAP modules to act as a copper bypass module, providing protected inline networks for copper ports.

Refer to the following sections about the rules and examples of how to configure the inline bypass solution on GigaVUE-OS TAP modules:

- Rules for Inline Bypass on TAP-HC0-G100C0 and TAP-HC1-G10040
- Example to Configure Inline Bypass on GigaVUE®HC Series Nodes

Related Topic:

- Refer to the *"Working with GigaVUE-OS HC TAP Modules"* chapter in the *GigaVUE Fabric Management Guide* for more information about the inline bypass solution for GigaVUE-OS TAP modules.

Rules for Inline Bypass on TAP-HC0-G100C0 and TAP-HC1-G10040

The TAP-HC1-G10040 module on the GigaVUE-HC1 features 4 pairs of relay-protected copper ports.

The protected port pairs on these modules can operate either as a copper TAP or as an inline network. Because the port pairs can have a dual role, the following rules apply:

- If a given pair of protected copper ports on the TAP-HC0-G100C0 or TAP-HC1-G10040 is not used for TAP-type arrangements (there is no port-pair configured between the protected ports and there are no out-of-band maps), the ports of such a pair are available for assigning the port type inline-network and for configuring an inline network.
- If a given pair of protected copper ports on the TAP-HC0-G100C0 or TAP-HC1-G10040 is used for TAP-type arrangements (there is a port-pair configured between the protected ports and, optionally, there are out-of-band maps), the ports of such a pair are blocked from assigning the port type inline-network and from configuring an inline-network.
- If a given pair of protected copper ports on the TAP-HC0-G100C0 or TAP-HC1-G10040 is used for protected inline bypass arrangements (the ports are assigned the port type inline-network), the ports are blocked from configuring a port-pair.
- Assigning a port type (either network or inline-network) to ports of a protected copper pair always affects both ports of the pair.

- Configuring an inline network using ports on the TAP-HC0-G100C0 or TAP-HC1-G10040 must enforce proper pairing of the net-a and net-b attributes of the inline network. This means that the ports selected as net-a and net-b must belong to the same pair of copper TAP ports.

Example to Configure Inline Bypass on GigaVUE® HC Series Nodes

The following example configures protected inline network ports on the TAP-HC1-G10040 module on the GigaVUE-HC1-Plus. It also configures inline tool ports, which are on a different module on the same GigaVUE-HC1-Plus.

- TAP-HC1-G10040 is a GigaVUE-HC1, GigaVUE-HC1-Plus and GigaVUE-HCT platform cards.

NOTE: The configuration steps in this example are similar to the configuration steps in the examples in the chapter Configure Inline Bypass Solutions, with the following exceptions:

- For protected inline networks, use the TAP-HC1-G10040 module instead of bypass combo modules on the GigaVUE-HC1-Plus.
- You need to configure inline network ports because they are not created automatically (as they are on bypass combo modules).
- You need to configure the inline network because it is not created automatically (as it is on bypass combo modules). In the following example, the inline network is named **inline_net_2_2_1**.

NOTE: On the GigaVUE-HC1, the configuration steps will be the same as in this example, but the network ports will be 1/3/g1..g8 (if the TAP-HC1-G10040 module is inserted in bay 3) and the tool ports will be 1/1/x1..x12 or 1/1g1..g4.

Step	Description	Command
1.	Configure ports on the TAP-HC0-G100C0 module as passive (in passive mode, relays are closed).	**(config) # port 2/2/x1..x2 params taptx passive**
2.	Configure inline network ports, port types (inline-network), and administratively enable inline network ports.	**(config) # port 2/2/x1 alias iN1** **(config) # port iN1 type inline-network** **(config) # port iN1 params admin enable** **(config) # port 2/2/x2 alias iN2** **(config) # port iN2 type inline-network** **(config) # port iN2 params admin enable**
3.	Configure inline network.	**(config) # inline-network alias inline_net_2_2_1 pair net-a iN1 and net-b iN2**

Step	Description	Command
4.	Configure inline tool ports, port types (inline-tool), and administratively enable inline tool ports.	**(config) # port 5/1/x3 alias iT1** **(config) # port iT1 type inline-tool** **(config) # port iT1 params admin enable** **(config) # port 5/1/x4 alias iT2** **(config) # port iT2 type inline-tool** **(config) # port iT2 params admin enable**
5.	Configure inline tool.	**(config) # inline-tool alias inline_tool pair tool-a iT1 and tool-b iT2**
6.	Configure map passall, from inline network to inline tool.	**(config) # map-passall alias inMap** (config map-passall alias **inMap**) # from inline_net_2_2_1 (config map-passall alias **inMap**) # to **inline_tool** (config map-passall alias **inMap)** # exit
7.	Configure the path of the traffic to inline tool.	**(config) # inline-network alias inline_net_2_2_1 traffic-path to-inline-tool**
8.	Disable physical bypass on the inline network alias. This opens the relays on the TAP-HC1-G10040 module.	**(config) # inline-network alias inline_net_2_2_1 physical-bypass disable**
9.	Display the configuration for this example.	**(config) # show inline-network** **(config) # show inline-tool** **(config) # show map**

Configure Flexible Inline Arrangements

Flexible inline arrangements is an approach to guide multiple inline traffic flows through a user-defined sequence of inline tools and inline tool groups. It uses the same software constructs as the existing inline bypass solution, such as inline network, inline tool, and inline tool group.

Flexible inline arrangements support physical protection based on the specialized hardware on BPS modules. It also supports both protected and unprotected inline network links. Refer to the *"Flexible Inline Arrangements"* chapter in the *GigaVUE Fabric Management Guide* for more information about the flexible inline arrangements solution.

IMPORTANT: Defining inline configurations through GigaVUE-FM is recommended.

- If you configure a flexible inline arrangement solution using the GigaVUE-OS CLI, you cannot view or manage it using GigaVUE-FM.
- If you modify a flexible inline arrangement solution using the GigaVUE-OS CLI, you cannot view the changes in GigaVUE-FM.

This section provides examples of how to configure the flexible inline arrangements solutions using the GigaVUE-OS CLI. They are presented in an order from simple to complex. Refer to the following:

- Example 1—Unprotected Flexible Inline, One Collector Map
- Example 1A—Unprotected Flexible Inline Netlag, One Collector Map
- Example 2—Unprotected Flexible Inline, Two Collector Maps
- Example 3—Protected Flexible Inline, Two Collector Maps
- Example 4—Unprotected Flexible Inline, Rule-Based Map
- Example 5—Unprotected Flexible Inline, Inline Tool Group
- Example 6—Unprotected Flexible Inline, Monitoring Mode
- Example 7—Protected Flexible Inline, Out-of-Band Copy

Example 1—Unprotected Flexible Inline, One Collector Map

Example 1 has one inline network, five inline tools, and a collector map that acts as a passall, sending all traffic through all tools.

The inline network alias is n0102, based on ports x1 and x2.

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), Advanced Persistent Threat (APT).

The inline tool aliases are t0708 to t1516, based on ports x7 to x16.

Use the following steps to configure Example 1:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x2 type inline-network** **(config) # port 1/3/x1..x2 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
4.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true

Step	Description	Command
	enable them. **Note**: The tag is optional. The default is **auto**, which automatically assigns tags.	(config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable **(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12** (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
5.	Configure collector map from inline network to inline tools in both directions, add user-defined tag, and enable map.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from n0102** **(config map alias FLEX1) # a-to-b t0708,t0910,t1112,t1314,t1516** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #**
6.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool**

Example 1A—Unprotected Flexible Inline Netlag, One Collector Map

Example 1 has one inline netlag, five inline tools, and a collector map that acts as a passall, sending all traffic through all tools.

The following two inline networks are configured in one inline netlag:

- inline network alias n0102, based on ports x1 and x2
- inilne network alias n0304, based on ports x3 and x4

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), Advanced Persistent Threat (APT).

The inline tool aliases are t0708 to t1516, based on ports x7 to x16.

Use the following steps to configure Example 1A:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x4 type inline-network** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline network.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2** **(config) # inline-network alias n0304 pair net-a 1/3/x3 and net-b 1/3/x4**
3.	Configure inline network lag.	**(config) # inline-netlag alias n0607** **network-list n0102,n0304**
4.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
5.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them. **Note**: The tag is optional. The default is **auto**, which automatically assigns tags.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable **(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12** (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true

Step	Description	Command
		(config) # inline-tool alias **t1516** enable
6.	Configure collector map from inline network to inline tools in both directions, add user-defined tag, and enable map.	**(config) # map alias FLEX1** (config map alias **FLEX1**) # type flexInline collector (config map alias FLEX1) # from n0607 FLEX1) # a-to-b **t0708,t0910,t1112,t1314,t1516** (config map alias **FLEX1**) # b-to-a reverse (config map alias **FLEX1**) # tag 100 (config map alias **FLEX1)** # enable (config map alias **FLEX1)** # exit **(config) #**
7.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool**

Example 2—Unprotected Flexible Inline, Two Collector Maps

Example 2 adds an inline network to Example 1. It has the same five inline tools, and adds a collector map for the second inline network. In the second collector map, two of the five tools are specified, sending traffic through those two tools, the first and the third tools in the sequence. The inline networks share these two tools.

The inline network aliases are n0102 to n0304, based on ports x1 to x4.

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), Advanced Persistent Threat (APT).

The inline tool aliases are t0708 to t1516, based on ports x7 to x16.

Figure 12Example 2 Inline Tool Sharing by Multiple Inline Flows illustrates Example 2. Traffic is only shown in one direction.

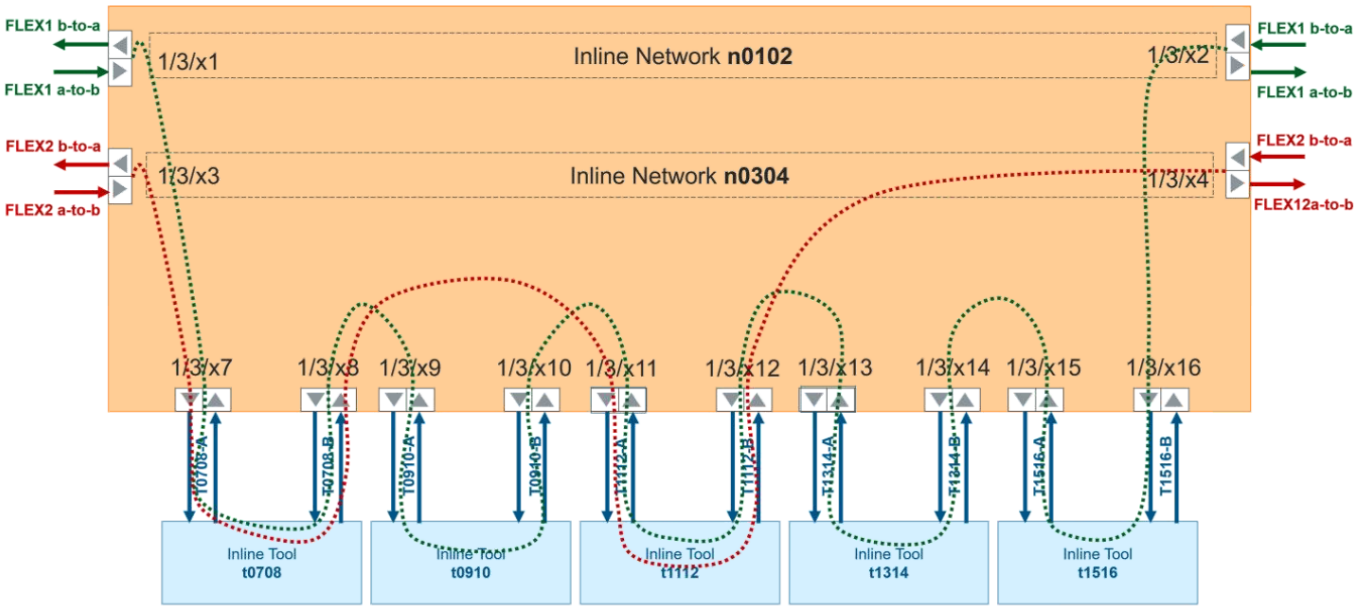


Figure 12 *Example 2 Inline Tool Sharing by Multiple Inline Flows*

Use the following steps to configure Example 2:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x4 type inline-network** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline networks.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2** **(config) # inline-network alias n0304 pair net-a 1/3/x3 and net-b 1/3/x4**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
4.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable

Step	Description	Command
		(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12 (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
5.	Configure collector maps from inline networks to inline tools in both directions, add user-defined tags, and enable maps. **NOTE:** The tag is optional. The default is **auto**, which automatically assigns tags.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from n0102** **(config map alias FLEX1) # a-to-b t0708,t0910,t1112,t1314,t1516** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #** **(config) # map alias FLEX2** **(config map alias FLEX2) # type flexInline collector** **(config map alias FLEX2) # from n0304** **(config map alias FLEX2) # a-to-b t0708,t1112** **(config map alias FLEX2) # b-to-a reverse** **(config map alias FLEX2) # tag 200** **(config map alias FLEX2) # enable** **(config map alias FLEX2) # exit** **(config) #**
6.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool** **(config) # inline-network alias n0304 traffic-path to-inline-tool**

Example 3—Protected Flexible Inline, Two Collector Maps

Example 3 is similar to Example 2 but with protected inline networks.

Protected inline networks are based on the pairs of ports associated with the physical protection switches located on the bypass combo modules. Unlike the unprotected examples, you do not need to configure inline network ports because they are created automatically, and you do not need to configure inline networks because they are also created automatically on bypass combo modules. The aliases of the default inline networks are: default_inline_net_1_1_1 and default_inline_net_1_1_2.

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), Advanced Persistent Threat (APT).

The inline tool aliases are t0708 to t1516, based on ports x7 to x16.

Use the following steps to configure Example 3:

Step	Description	Command
1.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
2.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable **(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12** (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
3.	Configure collector maps from inline networks to inline tools in both directions, add user-	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from default_inline_net_1_1_1**

Step	Description	Command
	defined tags, and enable maps. **Note**: The tag is optional. The default is **auto**, which automatically assigns tags.	**(config map alias FLEX1) # a-to-b t0708,t0910,t1112,t1314,t1516** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #** **(config) # map alias FLEX2** **(config map alias FLEX2) # type flexInline collector** **(config map alias FLEX2) # from default_inline_net_1_1_2** **(config map alias FLEX2) # a-to-b t0708,t1112** **(config map alias FLEX2) # b-to-a reverse** **(config map alias FLEX2) # tag 200** **(config map alias FLEX2) # enable** **(config map alias FLEX2) # exit** **(config) #**
4.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias default_inline_net_1_1_1 traffic-path to-inline-tool** **(config) # inline-network alias default_inline_net_1_1_2 traffic-path to-inline-tool**
5.	Disable physical bypass on the default inline networks.	**(config) # inline-network alias default_inline_net_1_1_1 physical-bypass disable** **(config) # inline-network alias default_inline_net_1_1_2 physical-bypass disable**

Example 4—Unprotected Flexible Inline, Rule-Based Map

Example 4 adds a rule-based map to Example 2. It has the same two inline networks, the same five inline tools, but adds a rule-based map from the first inline network. In the rule-based map, two of the five tools are specified, sending traffic through those two tools, which are the second (t0910) and the fourth (t1314) tools in the sequence.

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), Advanced Persistent Threat (APT).

Figure 13Example 4 Inline Tool Sharing by Multiple Inline Flows illustrates Example 4. Traffic is only shown in one direction.

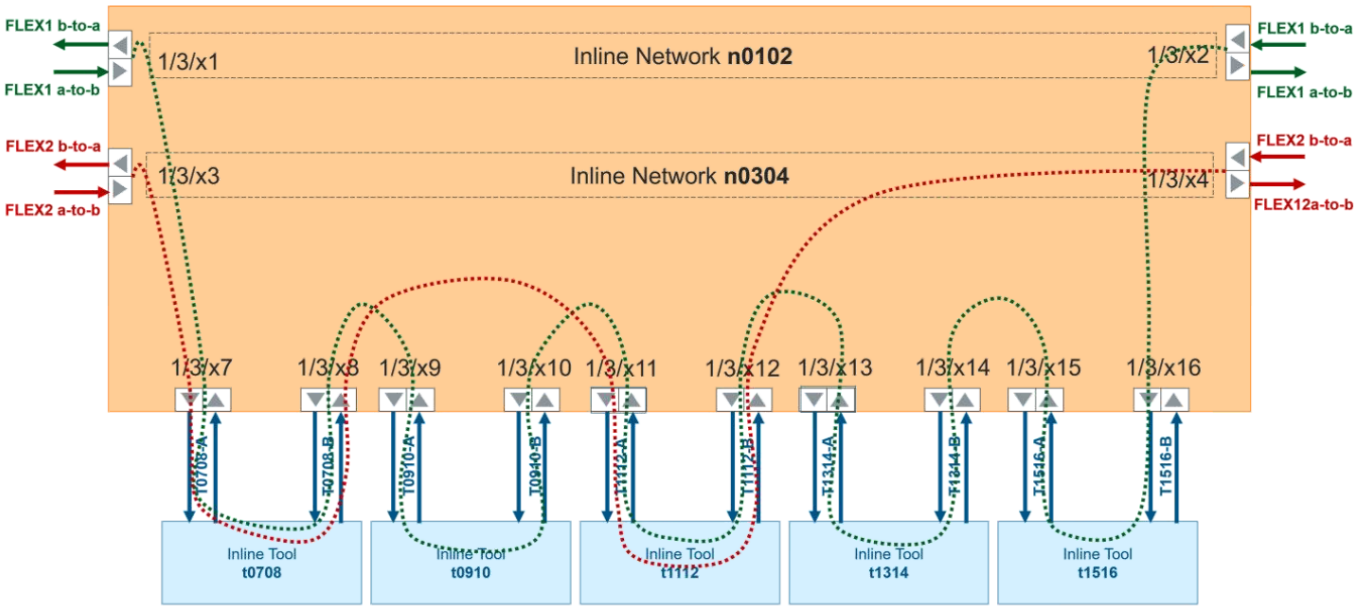


Figure 13 *Example 4 Inline Tool Sharing by Multiple Inline Flows*

Use the following steps to configure Example 4:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x4 type inline-network** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline networks.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2** **(config) # inline-network alias n0304 pair net-a 1/3/x3 and net-b 1/3/x4**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
4.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable

Step	Description	Command
		(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12 (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
5.	Configure maps from inline networks to inline tools in both directions, add user-defined tags, and enable maps. For the rule-based map, configure a rule (one rule only) to direct traffic to the tools. The rule can be based on any map rule criteria such as TCP port, IP subnet, or VLAN. **NOTE:** The tag is optional. The default is **auto**, which automatically assigns tags.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from n0102** **(config map alias FLEX1) # a-to-b t0708,t0910,t1112,t1314,t1516** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #** **(config) # map alias FLEX2** **(config map alias FLEX2) # type flexInline collector** **(config map alias FLEX2) # from n0304** **(config map alias FLEX2) # a-to-b t0708,t1112** **(config map alias FLEX2) # b-to-a reverse** **(config map alias FLEX2) # tag 200** **(config map alias FLEX2) # enable** **(config map alias FLEX2) # exit** **(config) #** **(config) # map alias FLEX3** **(config map alias FLEX3) # type flexInline byRule** **(config map alias FLEX3) # from n0102** **(config map alias FLEX3) # a-to-b t0910,t1314** **(config map alias FLEX3) # b-to-a reverse** **(config map alias FLEX3) # rule add pass ipver 4** **(config map alias FLEX3) # tag 300** **(config map alias FLEX3) # enable**

Step	Description	Command
		(config map alias FLEX3) # exit **(config) #**
6.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool** **(config) # inline-network alias n0304 traffic-path to-inline-tool**

Example 5—Unprotected Flexible Inline, Inline Tool Group

Example 5 adds an inline tool group to Example 4. It has the same two inline networks and five inline tools, but now the third, fourth, and fifth tools (t1112, t1314, and t1516) are in an inline tool group. The maps have been modified to direct traffic to the inline tool group.

For example, the inline tools can be Web Application Firewall (WAF), Intrusion Prevention System (IPS), while the Advanced Persistent Threat (APT) is the inline tool group.

The inline tool aliases are t0708 to t1516, based on ports x7 to x16. The inline tool group alias is ITG1.

Use the following steps to configure Example 5:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x4 type inline-network** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline networks.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2** **(config) # inline-network alias n0304 pair net-a 1/3/x3 and net-b 1/3/x4**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
4.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and**

Step	Description	Command
		tool-b 1/3/x10 (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable **(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12** (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
5.	Configure inline tool group and parameters. Enable it and then configure failover action.	**(config) # inline-tool-group alias ITG1** **(config inline-tool-group alias ITG1) # tool-list t1112,t1314,t1516** **(config inline-tool-group alias ITG1) # hash advanced** **(config inline-tool-group alias ITG1) # enable** **(config inline-tool-group alias ITG1) # failover-action tool-bypass** **(config inline-tool-group alias ITG1) # exit**
6.	Configure maps from inline networks to inline tools in both directions, add user-defined tags, and enable maps. For the rule-based map, configure a rule (one rule only) to direct traffic to the tools. The rule can be based on any map rule criteria such as TCP port, IP subnet, or VLAN. **Note**: The tag is optional. The default is **auto**, which automatically assigns tags.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from n0102** **(config map alias FLEX1) # a-to-b t0708,t0910,ITG1** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #** **(config) # map alias FLEX2** **(config map alias FLEX2) # type flexInline collector** **(config map alias FLEX2) # from n0304** **(config map alias FLEX2) # a-to-b t0708,ITG1** **(config map alias FLEX2) # b-to-a reverse** **(config map alias FLEX2) # tag 200** **(config map alias FLEX2) # enable**

Step	Description	Command
		(config map alias FLEX2) # exit **(config) #** **(config) # map alias FLEX3** **(config map alias FLEX3) # type flexInline byRule** **(config map alias FLEX3) # from n0102** **(config map alias FLEX3) # a-to-b ITG1** **(config map alias FLEX3) # b-to-a reverse** **(config map alias FLEX3) # rule add pass ipver 4** **(config map alias FLEX3) # tag 300** **(config map alias FLEX3) # enable** **(config map alias FLEX3) # exit** **(config) #**
7.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool** **(config) # inline-network alias n0304 traffic-path to-inline-tool**

Example 6—Unprotected Flexible Inline, Monitoring Mode

Example 6 adds a traffic path of monitoring for one inline tool to Example 4. It has the same two inline networks, the same five inline tools, and the same maps, but the flexible traffic path on the second inline tool is set to monitoring.

The monitoring mode is similar to bypass, but at the tool level. In a sequence of tools, you can select a separate tool to put into monitoring mode, in this case, it is the second tool, t0910.

Refer to Figure 14Example 6 Inline Tool Sharing by Multiple Inline Flows, the traffic returned from the B side of the network to t0910 will also be absorbed.

Figure 14Example 6 Inline Tool Sharing by Multiple Inline Flows illustrates Example 6. Traffic is only shown in one direction.

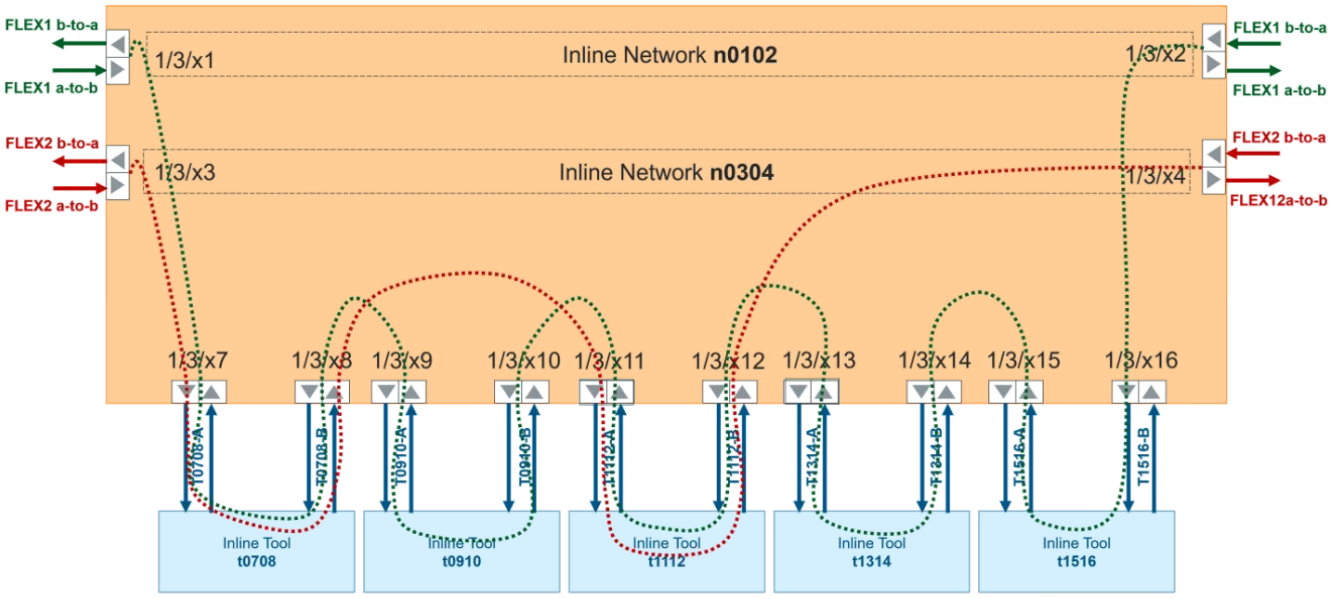


Figure 14 *Example 6 Inline Tool Sharing by Multiple Inline Flows*

Use the following steps to configure Example 6:

Step	Description	Command
1.	Configure inline network ports, port type (inline-network), and administratively enable inline network ports.	**(config) # port 1/3/x1..x4 type inline-network** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline networks.	**(config) # inline-network alias n0102 pair net-a 1/3/x1 and net-b 1/3/x2** **(config) # inline-network alias n0304 pair net-a 1/3/x3 and net-b 1/3/x4**
3.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x7..x16 type inline-tool** **(config) # port 1/3/x7..x16 params admin enable**
4.	Configure inline tools, specify that the inline tool is going to be shared by different sources, and enable them. On the second inline tool, specify a traffic path of monitoring.	**(config) # inline-tool alias t0708 pair tool-a 1/3/x7 and tool-b 1/3/x8** (config) # inline-tool alias **t0708** shared true (config) # inline-tool alias **t0708** enable **(config) # inline-tool alias t0910 pair tool-a 1/3/x9 and tool-b 1/3/x10** (config) # inline-tool alias **t0910 flex-traffic-path monitoring** (config) # inline-tool alias **t0910** shared true (config) # inline-tool alias **t0910** enable

Step	Description	Command
		(config) # inline-tool alias t1112 pair tool-a 1/3/x11 and tool-b 1/3/x12 (config) # inline-tool alias **t1112** shared true (config) # inline-tool alias **t1112** enable **(config) # inline-tool alias t1314 pair tool-a 1/3/x13 and tool-b 1/3/x14** (config) # inline-tool alias **t1314** shared true (config) # inline-tool alias **t1314** enable **(config) # inline-tool alias t1516 pair tool-a 1/3/x15 and tool-b 1/3/x16** (config) # inline-tool alias **t1516** shared true (config) # inline-tool alias **t1516** enable
5.	Configure maps from inline networks to inline tools in both directions, add user-defined tags, and enable maps. For the rule-based map, configure a rule (one rule only) to direct traffic to the tools. The rule can be based on any map rule criteria such as TCP port, IP subnet, or VLAN. **Note**: The tag is optional. The default is **auto**, which automatically assigns tags.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline collector** **(config map alias FLEX1) # from n0102** **(config map alias FLEX1) # a-to-b t0708,t0910,t1112,t1314,t1516** **(config map alias FLEX1) # b-to-a reverse** **(config map alias FLEX1) # tag 100** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #** **(config) # map alias FLEX2** **(config map alias FLEX2) # type flexInline collector** **(config map alias FLEX2) # from n0304** **(config map alias FLEX2) # a-to-b t0708,t1112** **(config map alias FLEX2) # b-to-a reverse** **(config map alias FLEX2) # tag 200** **(config map alias FLEX2) # enable** **(config map alias FLEX2) # exit** **(config) #** **(config) # map alias FLEX3** **(config map alias FLEX3) # type flexInline byRule** **(config map alias FLEX3) # from n0102** **(config map alias FLEX3) # a-to-b t0910,t1314** **(config map alias FLEX3) # b-to-a reverse** **(config map alias FLEX3) # rule add pass ipver 4** **(config map alias FLEX3) # tag 300** **(config map alias FLEX3) # enable**

Step	Description	Command
		(config map alias FLEX3) # exit **(config) #**
6.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias n0102 traffic-path to-inline-tool** **(config) # inline-network alias n0304 traffic-path to-inline-tool**

Example 7—Protected Flexible Inline, Out-of-Band Copy

Example 7 demonstrates a flexible inline map with OOB copy configuration as follows:

- an example of the source as a protected inline network and the destination as a hybrid port
- an example of the source as a tool member in the **a-to-b** list and the destination as a regular tool port
- an example of the source as a tool member in the **a-to-b** list and the destination as a GigaStream

Use the following steps to configure Example 7:

Step	Description	Command
1.	Configure inline tool ports, port type (inline-tool), and administratively enable inline tool ports.	**(config) # port 1/3/x1..x4 type inline-tool** **(config) # port 1/3/x1..x4 params admin enable**
2.	Configure inline tools, specify that the inline tool is going to be shared by different sources, specify heart-beat, and enable inline tools.	**(config) # inline-tool alias it1 pair tool-a 1/3/x1 and tool-b 1/3/x2** **(config) # inline-tool alias it1 shared true** **(config) # inline-tool alias it1 heart-beat** **(config) # inline-tool alias it1 enable** **(config) # inline-tool alias it2 pair tool-a 1/3/x3 and tool-b 1/3/x4** **(config) # inline-tool alias it2 shared true** **(config) # inline-tool alias it2 heart-beat** **(config) # inline-tool alias it2 enable**
3.	Configure hybrid port, port type (hybrid), and administratively enable hybrid port. The flexible inline map will configure out-of-band (OOB) traffic to this hybrid port.	**(config) # port 1/3/x19 type hybrid** **(config) # port 1/3/x19 params admin enable**

Step	Description	Command
4.	Configure regular tool ports, port type (tool), and administratively enable tool ports. The flexible inline map will configure out-of-band (OOB) traffic to a regular tool port. Two other tool ports will be used in a GigaStream.	**(config) # port 1/3/x20..x22 type tool** **(config) # port 1/3/x20..x22 params admin enable**
5.	Create a GigaStream using two of the regular tool ports.	**(config) # gigastream alias gs1 port-list 1/3/x21,1/3/x22**
6.	Configure the flexible inline map from the default inline network to inline tools in both directions, specify a rule, and a user-defined tag. Then configure out-of-band traffic as follows: ▪ from a protected inline network to a hybrid port using the same VLAN tag as the flexible inline map ▪ from the first tool member in the **a-to-b** list to a regular tool port without a VLAN tag. The tag can be configured to none, because traffic goes to a different destination, it1. ▪ from the second tool member in the **a-to-b** list to a GigaStream using the same VLAN tag as the flexible inline map Finally, enable the map.	**(config) # map alias FLEX1** **(config map alias FLEX1) # type flexInline byRule** **(config map alias FLEX1) # from default_inline_net_1_4_1** **(config map alias FLEX1) # rule add pass vlan 500** **(config map alias FLEX1) # a-to-b it1,it2** **(config map alias FLEX1) # b-to-a same** **(config map alias FLEX1) # tag 11** **(config map alias FLEX1) # oob-copy from default_inline_net_1_4_1 to 1/3/x19 tag as-inline** **(config map alias FLEX1) # oob-copy from it1 to 1/3/x20 tag none** **(config map alias FLEX1) # oob-copy from it2 to gs1 tag as-inline** **(config map alias FLEX1) # enable** **(config map alias FLEX1) # exit** **(config) #**
7.	Configure the path of the traffic to inline tools.	**(config) # inline-network alias default_inline_net_1_4_1 traffic-path to-inline-tool**
8.	Disable physical bypass on the default inline network.	**(config) # inline-network alias default_inline_net_1_4_1 physical-bypass disable**

NOTE: Starting in software version 6.4, you can use the tool/hybrid port which is part of a regular byRule or passall map with ingress VLAN tag enabled on the network port as an OOB-copy port of flexible inline maps as well. This update is compatible with all the GigaVUE HC Series, as well as the GigaVUE TA Series nodes GigaVUE-TA25 and GigaVUE-TA200, which support flexible inline solution.

Example 8—Flexible Inline Single Tag Configuration

When you configure inline maps with single VLAN tag, the map rules must have the same VLAN tag as configured in the **from** parameter.

The following is an example of a flexible inline single tag configuration.

```
map alias map1_in1_100_11
  type flexinline byRule
  rule add pass ipver 4 vlan 100
  from in1 vlan 100
  a-to-b it1_extTool,itg1
  b-to-a reverse
  tag 11
  oob-copy from in1 to 1/2/x1 tag original
  oob-copy from it1_extTool to 1/2/x1 tag original
  exit
```

Configure Inline SSL Decryption

Secure Sockets Layer (SSL) decryption for inline tools provides visibility into encrypted traffic. Inline SSL decryption delivers decrypted packets to tools that can be placed inline or out-of-band. The tools look into decrypted packets for threats, such as viruses or other malware.

Refer to the following sections for examples about how to configure SSL Decryption for inline tools for outbound and inbound deployments, and inline SSL session loging server:

- CLI Configuration Outbound Example
- CLI Configuration Inbound Example
- Configure an Inline SSL Session Logging Server Using CLI

Related Topics:

- Refer to the *"Inline SSL Decryption"* chapter in the *GigaVUE Fabric Management Guide* for more information about the SSL decryption for inline and out-of-band tools.

CLI Configuration Outbound Example

The following is an example of SSL Decryption for inline tools configuration for an outbound deployment using the CLI. Any CLI command or option that does not have to be configured because it has default values is not included.

Step	Description	Command
1.	Configure a keychain password. **Note**: The keychain password	**(config) # apps inline-ssl keychain password**Creating a new password for ssl keychain:Password: ******* Confirm: *******

Step	Description	Command
	must be configured before installing certificates and keys. If the key has a passphrase, in order to install it, the keychain password and the passphrase must match.	
2.	(Optional) Configure trust store, which installs trusted certificate authority (CA) for server certificate validation.	**(config) # apps inline-ssl trust-store fetch http://1.1.1.1/mitm/my_trust_store.pem**
3.	Configure the Man-in-the-Middle (MitM) primary CA private key and certificate. Then bind them to the primary CA. The primary CA re-signs certificates for servers that present a valid certificate. **Note**: The secondary CA private key and certificate can be configured, but is optional.	**(config) # apps keystore rsa primary private-key download url http://1.1.1.1/mitm/primary_ca.key** **(config) # apps keystore rsa primary certificate download url http://1.1.1.1/mitm/primary_ca.cert** **(config) # apps inline-ssl signing rsa for primary key primary**
4.	Configure an inline SSL profile. The profile specifies policy configuration, such as certificate handling and actions to take for the profile. **Note**: This sample profile is a decrypt all profile, meaning that all SSL traffic is decrypted. From a compliance point of view, check the necessary IT compliance criteria of your organization. The default value for **tcp syn-retries** is 3. The default value for **tool fail-action** is bypass-tool.	**(config) # apps inline-ssl profile alias sslprofile** **(config apps inline-ssl profile alias sslprofile) # certificate expired drop** **(config apps inline-ssl profile alias sslprofile) # certificate invalid decrypt** **(config apps inline-ssl profile alias sslprofile) # certificate revocation crl disable** **(config apps inline-ssl profile alias sslprofile) # certificate revocation ocsp disable** **(config apps inline-ssl profile alias sslprofile) # certificate self-signed decrypt** **(config apps inline-ssl profile alias sslprofile) # certificate unknown-ca decrypt** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp inactive-timeout 5** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp portmap default-out-port disable** **(config apps inline-ssl profile alias sslprofile) # decrypt tool-bypass disable** **(config apps inline-ssl profile alias sslprofile) # default-action decrypt** **(config apps inline-ssl profile alias sslprofile) #**

Step	Description	Command
		no-decrypt tool-bypass disable **(config apps inline-ssl profile alias sslprofile) # url-cache miss action decrypt** **(config apps inline-ssl profile alias sslprofile) # exit** **(config) #**
5.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gs1 port-list 2/1/e1**
6.	Configure the GigaSMART inline SSL operation, specify the profile, and assign the GigaSMART operation to the GigaSMART group.	**(config) # gsop alias issl inline-ssl sslprofile port-list gs1**
7.	Configure a virtual port and assign it to the same GigaSMART group. Then configure a failover action on the virtual port.	**(config) # vport alias vport1 gsgroup gs1** **(config) # vport alias vport1 failover-action vport-bypass**
8.	Configure tool ports with type tool, and administratively enable tool ports. These ports are optional, for the out-of-band map.	**(config) # port 10/1/x13..x16 type tool** **(config) # port 10/1/x13..x16 params admin enable**
9.	Configure inline network ports with type inline-network, and administratively enable inline network ports.	**(config) # port 2/2/x11..x12 type inline-network** **(config) # port 2/2/x11..x12 params admin enable** **(config) # port 2/2/x13..x14 type inline-network** **(config) # port 2/2/x13..x14 params admin enable** **(config) # port 2/2/x17..x18 type inline-network** **(config) # port 2/2/x17..x18 params admin enable** **(config) # port 2/2/x19..x20 type inline-network** **(config) # port 2/2/x19..x20 params admin enable**
10.	Configure inline networks. In this example, the inline networks are unprotected. **Note**: Only one inline network needs to be specified. The others are optional.	**(config) # inline-network alias inline-net1 pair net-a 2/2/x19 and net-b 2/2/x20** **(config) # inline-network alias inline-net2 pair net-a 2/2/x13 and net-b 2/2/x14** **(config) # inline-network alias inline-net3 pair net-a 2/2/x17 and net-b 2/2/x18** **(config) # inline-network alias inline-net4 pair net-a 2/2/x11 and net-b 2/2/x12**
11.	(Optional) Configure inline	**(config) # inline-network-group alias ing1**

Step	Description	Command
	network group. This example has four inline networks in an inline network group. **Note**: If only one inline network is specified, the inline network group is optional.	**(config inline-network-group alias ing1) # network-list inline-net1,inline-net2, inline-net3, inline-net4** **(config inline-network-group alias ing1) # exit** **(config) #**
12.	Configure inline tool ports with type inline-tool, and administratively enable inline tool ports.	**(config) # port 2/2/x3..x4 type inline-tool** **(config) # port 2/2/x3..x4 params admin enable** **(config) # port 2/2/x5..x6 type inline-tool** **(config) # port 2/2/x5..x6 params admin enable**
13.	Configure inline tools and enable them. **Note**: Only one inline tool needs to be specified. The others are optional.	**(config) # inline-tool alias it1 pair tool-a 2/2/x3 and tool-b 2/2/x4** **(config) # inline-tool alias it2 pair tool-a 2/2/x5 and tool-b 2/2/x6** it1 enable (config) # inline-tool alias **it2** enable
14.	Enable default heartbeat.	**(config) # inline-tool alias it1 heart-beat** **(config) # inline-tool alias it2 heart-beat**
15.	Specify that inline tools are going to be shared by different sources. When shared is enabled (true), the inline tool can receive traffic from multiple sources (inline networks and GigaSMART).	**(config) # inline-tool alias it1 shared true** **(config) # inline-tool alias it2 shared true**
16.	(Optional) Configure inline tool group and parameters. Then enable inline tool group. This example has two inline tools in an inline tool group. **Note**: If only one inline tool is specified, the inline tool group is optional.	**(config) # inline-tool-group alias itg1** **(config inline-tool-group alias itg1) # tool-list it1,it2** **(config inline-tool-group alias itg1) # minimum-group-healthy-size 2** **(config inline-tool-group alias itg1) # enable** **(config inline-tool-group alias itg1) # exit** **(config) #**
17.	Configure first level inline SSL map. This map has a rule that passes TCP traffic, and then directs it from the inline network group to a virtual port (and to GigaSMART). This map (and the next) is for traffic that needs to be decrypted	**(config) # map alias inline-issl-L1map1** **(config map alias inline-issl-L1map1) # rule add pass protocol tcp** **(config map alias inline-issl-L1map1) # to vport1** **(config map alias inline-issl-L1map1) # from ing1** **(config map alias inline-issl-L1map1) # exit** **(config) #**

Step	Description	Command
	so the tools can inspect it, such as HTTPS traffic. The map type and subtype are determined by the **from** and **to** parameters (inLineFirstLevel, ingresstovp).	
18.	Configure classic inline map. This map directs traffic from the inline network group to the inline tool group, using a specified rule. It has the same **from** port as the first level inline SSL map and the same **to** port as the second level inline SSL map. This map is for traffic that does not need to be decrypted for the tools to inspect it, such as non-HTTPS traffic or UDP traffic.	**(config) # map alias inline-bypass1** **(config map alias inline-bypass1) # rule add pass ipver 4** **(config map alias inline-bypass1) # to itg1** **(config map alias inline-bypass1) # from ing1** **(config map alias inline-bypass1) # exit** **(config) #**
19.	Configure a shared collector for any unmatched traffic including non-TCP traffic, which is directed to bypass.	**(config) # map-scollector alias isslscoll** **(config map alias isslscoll) # from ing1** **(config map alias isslscoll) # collector bypass** **(config map alias isslscoll) # exit** **(config) #**
20.	Configure second level inline SSL map. This map directs traffic from the virtual port, uses the inline SSL GigaSMART operation, and sends traffic to the inline tool group. The map type and subtype are determined by the **from** and **to** parameters (inLineSecondLevel, egressfromvp).	**(config) # map alias inline-issl-L2map1** **(config map alias inline-issl-L2map1) # use gsop issl** **(config map alias inline-issl-L2map1) # to itg1** **(config map alias inline-issl-L2map1) # from vport1** **(config map alias inline-issl-L2map1) # exit** **(config) #**
21.	(Optional) Configure an out-of-band map to a single tool port, multiple tool ports, single hybrid port, GigaStream, or port group with tool or hybrid ports, or combination of these. This example has multiple tool ports.	**(config) # map alias gs-oob** **(config map alias gs-oob) # use gsop issl** **(config map alias gs-oob) # to 10/1/x13..x16** **(config map alias gs-oob) # from vport1** **(config map alias gs-oob) # exit** **(config) #**

Step	Description	Command
22.	Configure the path of the traffic to inline tool on the inline networks.	**(config) # inline-network alias inline-net1 traffic-path to-inline-tool** **(config) # inline-network alias inline-net2 traffic-path to-inline-tool** **(config) # inline-network alias inline-net3 traffic-path to-inline-tool** **(config) # inline-network alias inline-net4 traffic-path to-inline-tool**
23.	(Optional) If the inline networks are protected, disable physical bypass.	**(config) # inline-network alias inline-net1 physical-bypass disable** **(config) # inline-network alias inline-net2 physical-bypass disable** **(config) # inline-network alias inline-net3 physical-bypass disable** **(config) # inline-network alias inline-net4 physical-bypass disable**

CLI Configuration Inbound Example

For inspecting inbound SSL sessions, the server's key pair must be installed in the key store and the inline SSL profile must have the corresponding key map configured.

The primary MitM CA is not mandatory for an inbound deployment.

Most of the steps for an inbound deployment are the same as the outbound deployment, with the following exceptions:

- Skip step 1 and step 2. Instead, use the following CLI command to download private keys:

  (config) # apps keystore rsa server_chain_001 pkcs12 download url <URL>

  The supported formats for <URL> are HTTP, FTP, TFTP, SCP, and SFTP.

  For example:

  (config) # apps keystore rsa server_chain_001 pkcs12 download url sftp://test:mytest@10.10.10.10/home/test/ssldecrypt/keys/srv1k.pfx

- When configuring the inline SSL profile in step 4, include the following CLI command to create a key map entry:

  (config apps inline-ssl profile alias sslprofile) # keymap add server server_chain_001 key server_chain_001

Configure an Inline SSL Session Logging Server Using CLI

You can configure an inline SSL session logging server to store the logged events that are generated when there are any changes made to the devices. You can specify the type of events that must be logged in to the server.

The following table provides a mapping of the severity, log level and its description:

Severity	Log Level	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical condition
3	Error	Error condition
4	Warning	Warning condition
5	Notice	Normal but significant condition
6	Informational	Informational message
7	Debug	Debug message

The logged events are stored in the Common Event Format (CEF) as follows:

<SYSLOG_HEADER> <Timestamp> <hostname:engine> CEF:0|Gigamon|<Device Model>|< GigaVUE-OS OS Version>|<Event ID>|<Event name>|<Severity>|[Extension]

Here is an example of a logged event:

Thu Jun 14 15:50:16 2018 hostname:hc3_test:1/1/e1CEF:0|Gigamon|HC3|5.5.0|102|SESSION_ DECRYPT|6|src=126.1.0.20dst=126.1.0.10 spt=34267 dpt=443 dhost=example.comcs1Label=Certificate Subject cs1=C\=US, ST\=CA, L\=Santa Clara,CN=*.example.com cs2Label=Cipher Suite cs2=DHE-RSA-AES128-GCM-SHA256

You can view and track these logs to troubleshoot system issues, maintain audit trails, and for compliance purpose.

To configure an inline SSL session logging server:

Step	Description	Command
1.	Configure an IP interface and attach a GigaSMART group.	**(config) # ip interface <port alias> attach <tool_ port_id> ip <IP address> <netmask | mask length> gateway <gateway IP address> gsggroup add <GigaSMART group alias>**

Step	Description	Command
2.	Configure the session log levels under the GigaSMART parameters (gsparams). **Note**: If you set the session log level as None, the logs will not be sent to the inline SSL session logging server.	**HC3 (config) # gsparams gsgroup <alias> session logging level <err|warning|notice|info|debug|none>**
3.	Add the inline SSL session logging server details under the GigaSMART parameters (gsparams). **Note**: You can configure only one inline SSL session logging server.	**HC1 (config) # gsparams gsgroup <alias> session logging add remote-ip <syslog_ip | ipv6 <syslog_ipv6>> portdst <port> interface <ip_interface | ipv6_interface>**

Use the following CLI command to delete the configured inline SSL session logging server:

HC1 (config) # gsparams gsgroup <alias> session logging delete remote-ip <syslog_ip | <syslog_ipv6>>

NOTE: IPV6 traffic decryption is supported only for GEN 3 cards. Refer to the GigaVUE-HC1 Hardware Installation Guide and GigaVUE-HC3 Hardware Installation Guide for the list of GEN 3 card numbers.

Configure GigaSMART Operations

Use the **gsop** command to create a GigaSMART operation. GigaSMART operations consist of a name and a supported combination of the available GigaSMART applications you have licensed.

GigaSMART Operations – Example

The following procedure summarizes the major steps in creating and using a GigaSMART operation.

Summary	Command
Start by using the gsgroup command to create a GigaSMART group – a collection of one or more internal GigaSMART engine ports available in a given chassis. GigaSMART groups are used to process GigaSMART operations – each GSOP you create is assigned to a GigaSMART group. In this example, we have created a GigaSMART group called GS1 using virtual port e1 on the GigaSMART-HC0	**(config) # gsgroup alias GS1 port-list 16/2/e1**

Summary	Command
line card in slot 2 of box 16 (16/2/e1).	
Next, you can create a GigaSMART operation – a combination of actions that can be used in a map – and assign it to a GigaSMART group for processing. In this example, we have created a GigaSMART operation called tcpmask that will overwrite 16 bytes of packet data starting 64 bytes after the end of the TCP header using a hexadecimal ee pattern. We have also assigned it to the GS1 GigaSMART group we created in the first step.	**(config) # gsop alias tcpmask masking protocol tcp offset 64 pattern ee length 16 port-list GS1**
Once you have set up a GigaSMART operation, you can include it as part of a map with the use gsop command in the map prefix mode. In this example, the tcpmask GigaSMART operation is combined with an IP Version pass rule so that all IPv4 traffic processed is masked according to the GSOP we created in the previous step. If you are not sure which GigaSMART operation you want to use, use the ? mark after the use gsop command to display the operations you have already configured. **Note**: These commands show how to include GigaSMART operations in the CLI's map prefix mode. Here, we have created a map named gsmap that will forward IPv4 traffic from network ports 16/3/x7..x12 to tool port 16/3/x1. The traffic will be masked using the tcpmask GigaSMART operation we created in Step 2	**(config) # map alias gsmap** **(config map alias gsmap) # type regular byRule** **(config map alias gsmap) # from 16/3/x7..x12** **(config map alias gsmap) # use gsop tcpmask** **(config map alias gsmap) # to 16/3/x1** **(config map alias gsmap) # rule add pass ipver 4** **(config map alias gsmap) # exit**

The configuration examples of each GigaSMART operation is described in the following sections:

- Configure GigaSMART Masking
- Configure Packet Slicing
- GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel)
- GigaSMART IP Encapsulation (GigaSMART Tunnel)
- GigaSMART IP Encapsulation (GigaSMART Tunnel)
- IP Fragmentation and Reassembly on L2GRE and GMIP Tunnels
- Tunnel Health Checks
- GigaSMART ERSPAN Tunnel Decapsulation
- GigaSMART VXLAN Tunnel Decapsulation
- GigaSMART Custom Tunnel Decapsulation
- GigaSMART Header Addition
- GigaSMART De-duplication

- GigaSMART Header Stripping
- GigaSMART GTP Correlation
- GigaSMART GTP Whitelisting and GTP Flow Sampling Examples
- GigaSMART GTP Overlap Flow Sampling Maps
- GigaSMART SIP/RTP Correlation
- GigaSMART GTP Scaling
- GigaSMART GTP Stateful Session Recovery
- GigaSMART FlowVUE
- GigaSMART Application Session Filtering (ASF) and Buffer ASF
- GigaSMART NetFlow Generation
- GigaSMART Load Balancing
- GigaSMART MPLS Traffic Performance Enhancement
- GigaSMART SSL Decryption for Out-of-Band Tools
- Entrust nShield Hardware Security Module (HSM)
- GigaSMART FlowVUE
- GigaSMART Application Session Filtering (ASF) and Buffer ASF
- GigaSMART NetFlow Generation

Related Topics

- Refer to the "*How to Combine GigaSMART Operations*" section in the *GigaVUE Fabric Management Guide* for details on supported combinations of GigaSMART operations.
- Refer to the "*Order of GigaSMART Operations*" section in the *GigaVUE Fabric Management Guide* for information on the order in which GigaSMART components are applied in a single operation.
- Refer to the *"GigaSMART Operations Statistics Definitions"* section in the *GigaVUE Fabric Management Guide* for information about GigaSMART operations statistics.
- Refer to gsop in the reference section for details on the syntax of the GigaSMART operations CLI command.

GigaSMART Operations - Limitations

- The interface of the GigaSMART card will go down and traffic drops will become visible when the traffic rate exceeds 80% of the line rate.

Configure GigaSMART Masking

GigaSMART operations with a **masking** argument will write over a specific portion of a packet with a specified one-byte pattern.

Refer to the *"GigaSMART Masking"* section in the *GigaVUE Fabric Management Guide* for details about the masking operations.

Following is an example of how you can configure GigaSMART Masking:

Summary	Command
This example creates a GigaSMART masking operation named **tunnel_mask**. This example starts masking six bytes after the end of the TCP layer in the GTP-encapsulated packet and continues for 150 bytes, writing over the existing data with an FF pattern.	**(config) # gsop alias tunnel_mask masking protocol gtp-tcp offset 6 pattern FF length 150 port-list GS1**
This example creates a GigaSMART masking operation named **Mask_FIX**. This example uses a static masking offset of 148 bytes and continues for the next 81 bytes, writing over the existing data with an **FF** pattern. This GigaSMART operation is assigned to the GigaSMART group with the alias of GS2. This example simulates how to mask a FIX (Financial Information eXchange) packet so that generic information is preserved at the start and end of the FIX data portion of the packet while private information within is masked. This example does not include the optional GigaSMART Trailer.	**(config) # gsop alias Mask_FIX masking protocol none offset 148 pattern 0xFF length 81 port-list GS2**

Configure Packet Slicing

GigaSMART operations with a slicing component truncate packets after either a specified header/layer and offset (a relative offset) or at a specific offset. Slicing operations are typically configured to preserve specific packet header information, allowing effective network analysis without the overhead of storing full packet data.

Refer to the *"GigaSMART Packet Slicing"* section in the *GigaVUE Fabric Management Guide* for details about the packet slicing operations.

Following is an example of how you can configure GigaSMART Packet Slicing:

Summary	Command
This example creates a GigaSMART slicing operation named **IPv6_Headers**. This operation truncates all packet data starting four bytes after the IPv6 header. The sliced packet would include the DLC, IPv6, and TCP headers, which are often enough for analysis needs.	**(config) # gsop alias IPv6_Headers slicing protocol ipv6 offset 4 port-list GS2**

GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel)

Use GigaSMART encapsulation and decapsulation operations to send traffic arriving on one GigaSMART-enabled node over the Internet to a second GigaSMART-enabled node. There, the traffic is decapsulated and made available to local tool ports.

Refer to the *"GigaSMART IP Encapsulation/Decapsulation (GigaSMART Tunnel)"* section in the *GigaVUE Fabric Management Guide* for details about GigaSMART IP Encapsulation/Decapsulation details.

The following example describes how to configure the sending end of the tunnel for the physical devices in different location.

Configure the Sending End of the Tunnel: GigaVUE-HC1 in Reno

The GigaVUE-HC1 in this location has an IP interface configured on tool port 1/1/g1 with an IP address of 11.1.9.75. Maps to this port that use a tunnel encapsulation GigaSMART operation can send data over the Internet. The following table summarizes the commands necessary to configure the sending end of the tunnel in the CLI:

Task	Commands
Start by designating port 1/1/g1 as a tool port.	**(config) # port 1/1/g1 type tool**
Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport1 port-list 1/1/e1**
Use the **ip interfaces** command to set up the network parameters for 1/1/g1. This command sets the IP address, subnet mask, default gateway, and MTU for the IP interface with tool port on port 1/1/g1. Notice that the GigaSMART group in this example has the alias **gsport1**.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g1** **(config ip interface alias test) # ip address 11.1.9.75/29** **(config ip interface alias test) # gw 11.1.9.1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsport1** **(config ip interface alias test) # exit**
Now, create a tunnel encapsulation GigaSMART operation (**tunnelenc**) that will send traffic to IP address 21.2.9.75 on destination UDP port 10000 from source port 5000. The operation has the alias **tunnelenc**.	**config) # gsop alias tunnelenc tunnel-encap type gmip portsrc 5000 portdst 10000 ipdst 21.2.9.75 prec 1 port-list gsport1**
Once we have our tunnel encapsulation operation, we can include it as part of a map rule. This map rule matches IPv4 packets and sends them to 21.2.9.75:10000 (the socket specified by the GigaSMART operation named **tunnelencap** we created in the previous step).	**(config) # map alias tunnelencap** **(config map alias tunnelencap) # type regular byRule** **(config map alias tunnelencap) # use gsop tunnelenc** **(config map alias tunnelencap) # to 1/1/g1** **(config map alias tunnelencap) # from 1/1/x3** **(config map alias tunnelencap) # rule add pass ipver 4** **(config map alias tunnelencap) # exit**

Configure the Receiving End of the Tunnel: GigaVUE-HC3 with GigaSMART in San Francisco

Now we need to configure the receiving end of the tunnel with an IP interface associated with the network port. The GigaVUE-HC3 in this location will have an IP interface associated with the network port configured on network port 5/1/c2 with an IP address of 21.2.9.75 and a GigaSMART decapsulation operation that listens on UDP port 10000.

The following table summarizes the commands necessary to configure the receiving end of the tunnel in the CLI:

Task	Commands
Start by designating port 5/1/c2 as a network port.	**(config) # port 5/1/c2 type network**
Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport5 port-list 1/1/e1**
Use the **ip interfaces** command to set up the network parameters for 5/1/c2. This command sets the IP address, subnet mask, default gateway, and MTU for the IP interface associated with the network port on port 5/1/c2. Note that this port uses the same IP address to which the GSOP in Reno is configured to send data (21.2.9.75).	**(config) # ip interface alias test** **(config ip interface alias test) # attach 5/1/c2** **(config ip interface alias test) # ip address 21.2.9.75/29** **(config ip interface alias test) # gw 21.2.9.1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsport5** **(config ip interface alias test) # exit**
Now, create a tunnel decapsulation GigaSMART operation (**tunnel-decap**) that will decapsulate traffic received on UDP port 10000. Recall that we configured the sending end of the tunnel to send to that UDP port. The operation has the alias **hc-decap1**.	**(config) # gsop alias hc-decap1 tunnel-decap type gmip portdst 10000 port-list gsport5**
Once we have our tunnel decapsulation operation, we can include it as part of a map rule. This map decapsulates all traffic arriving at 5/1/c2 from IP address 21.2.9.25 (the start of the tunnel) and sends it to port 1/1/c2. This is a tool port on the chassis with box ID 1 in this cluster.	**(config) # map alias decapper1** **(config map alias decapper1) # type regular byRule** **(config map alias decapper1) # use gsop hc-decap1** **(config map alias decapper1) # to 1/1/c2** **(config map alias decapper1) # from 5/1/c2** **(config map alias decapper1) # rule add pass ipsrc 11.1.9.75 255.255.255.0** **(config map alias decapper1) # exit**

Configure the Receiving End of the Tunnel: GigaVUE-OS® HC Series with GigaSMART in Melbourne

Now we need to configure the receiving end of the tunnel with an IP interface associated with the network port. TheGigaVUE HC Series in this location will have an IP interface associated with the network port configured on network port 1/1/3 with an IP address of 10.150.68.222 and a GigaSMART decapsulation operation that listens on UDP port 10000.

The following table summarizes the commands necessary to configure the receiving end of the tunnel in the CLI:

Task	Commands
Start by designating port 1/1/x3 as a network port associated with an IP interface, configuring its IP profile, and add the required GigaSMART group to the IP interface. This command sets the IP address, subnet mask, default gateway, and MTU for the IP interface with tool port on port 1/1/x3.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/x3** **(config ip interface alias test) # ip address 10.150.68.222 /32** **(config ip interface alias test) # gw 10.150.68.1** **(config ip interface alias test) #** mtu 9400 **(config ip interface alias test) # gsgroup add gs2** **(config ip interface alias test) # exit**
Now, create an IP decapsulation GigaSMART operation (**gv_ipdecap**) that will decapsulate traffic received on UDP port 10000. Recall that we configured the sending end of the tunnel to send to that UDP port. The operation has the alias **gv_ipdecap**. Note that this operation uses the same GigaSMART group (GS2) as the network port associated with the IP interface we set up in the first step.	**(config) # gsop alias gv_ipdecap tunnel-decap type gmip portdst 10000 port-list GS2**
Once we have our IP decapsulation operation, we can include it as part of a map. ▪ The map alias command opens the map prefix mode with a map named **decapper**. ▪ The **from** command specifies the ingress ports for this map. ▪ The **use gsop** command applies the gv_ipdecap GigaSMART operation to all packets matching the rules in the map, decapsulating them from the tunnel.	**(config) # map alias decapper** (config map alias decapper) # type regular byRule (config map alias decapper) # from 1/1/x3 (config map alias decapper) # use gsop gv_ipdecap (config map alias decapper) # to 1/1/x11 (config map alias decapper) # rule add pass ipsrc 10.10.10.10 /32 (config map alias decapper) # exit (config) #

Task	Commands
▪ The **to** command specifies where matching packets will be sent (tool port 1/1/x11). ▪ The rule add pass command specifies that packets arriving on this port with an IP Source address of 10.10.10.10 /32 will be processed by the gv_ipdecap GSOP and sent to tool port 1/1/x11.	

GigaSMART IP Encapsulation (GigaSMART Tunnel)

GigaSMART-enabled nodes with the Advanced Tunneling license installed can encapsulate traffic and send it through a GigaSMART tunnel to a destination GigaSMART-enabled node.

Refer to the "*GigaSMART IP Encapsulation (GigaSMART Tunnel)*" section in the *GigaVUE Fabric Management Guide* for details.

GigaSMART Layer 2 GRE Tunnel Encapsulation/Decapsulation

GigaSMART-enabled nodes with the Advanced Tunneling license installed can encapsulate traffic and send it through a GigaSMART tunnel to a destination GigaSMART-enabled node.

Use GigaSMART Layer 2 (L2) Generic Routing Encapsulation (GRE) tunnel encapsulation to send traffic from one GigaSMART node over the Internet to a second GigaSMART node using L2GRE encapsulation. Use GigaSMART L2GRE tunnel decapsulation at the second GigaSMART node to decapsulate the traffic before sending it to local tool ports.

GigaSMART Layer 2 GRE tunnel encapsulation/decapsulation provides the following:

- L2GRE tunnel initiation and encapsulation on the tool port at the sending end of the tunnel (for example, at a remote site)
- L2GRE tunnel termination and decapsulation on the network port at the receiving end of the tunnel (for example, at a main office site)

Refer to the following configuration examples:

- Example 1 – GigaSMART L2GRE Tunnel Encapsulation
- Example 2 – GigaSMART L2GRE Tunnel Encap Stateful LB
- Example 3 – GigaSMART L2GRE Tunnel Encap Stateless LB
- Example 4 – GigaSMART L2GRE Tunnel Decapsulation
- Example 5 – GigaSMART L2GRE IPv6 Tunnel Encap/Decap

Example 1 – GigaSMART L2GRE Tunnel Encapsulation

In this example, an IP interface is configured on the tool port. A GigaSMART operation for tunnel encapsulation is configured to encapsulate the filtered packets. A map is configured that uses the L2GRE tunnel encapsulation GigaSMART operation, which sends packets from the remote site over the Internet to the main office using the IP interface with tool port.

Step	Description	Command
1.	Configure a tool type of port and a network type of port.	**(config) # port 1/1/x1 type tool** **(config) # port 1/1/x2 type network**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport1 port-list 1/1/e1**
3.	Configure the IP interface with an IP address, subnet mask, default gateway, and MTU setting. Assign it to the GigaSMART group.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/x1** **(config ip interface alias test) # ip address 1.1.1.1 /29** **(config ip interface alias test) # gw 1.1.1.2** **(config ip interface alias test) #** mtu 9400 **(config ip interface alias test) # gsgroup add gsport1** **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation for tunnel encapsulation and assign it to the GigaSMART group. The tunnel encapsulation settings include the IP address (IPv4) of the IP interface on the destination GigaSMART node and the GRE key that identifies the source of the tunnel.	**(config) # gsop alias gsop1 tunnel-encap type l2gre ipdst 4.4.4.4 key 12314 port-list gsport1**
5.	Create a map using the tunnel encapsulation GigaSMART operation, with packets coming from the network port and being sent to the Internet through the tool port.	**(config) # map alias tun_encap** **(config map alias tun_encap) # type regular byRule** **(config map alias tun_encap) # use gsop tunnelencap** **(config map alias tun_encap) # rule add pass ipver 4** **(config map alias tun_encap) # from 1/1/x2** **(config map alias tun_encap) # to 1/1/x1** **(config map alias tun_encap) # exit** **(config) #**

Step	Description	Command
6.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show ip interfaces** **(config) # show gsop** **(config) # show map**

Example 2 – GigaSMART L2GRE Tunnel Encap Stateful LB

Example 2 configures stateful load balancing of tunnel traffic to three tunnel endpoints based on a metric. Each tunnel endpoint is assigned a weight.

Step	Description	Command
1.	Configure a tool type of port and a network type of port.	**(config) # port 1/3/x2 type tool** **(config) # port 1/3/x1 type network**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport1 port-list 1/1/e1**
3.	Configure the IP interface with an IP address, subnet mask, default gateway, and MTU setting. Assign it to the GigaSMART group.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/3/x2** **(config ip interface alias test) # ip address 1.1.1.1 /29** **(config ip interface alias test) # gw 1.1.1.100** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsport1** **(config ip interface alias test) # exit**
4.	Configure tunnel endpoints. The alias is optional.	**(config) # tunnel-endpoint te-id te1 alias tunnel_ endpoint1** **(config) # tunnel-endpoint te-id te1 type remote ip-address 1.1.1.200** **(config) # tunnel-endpoint te-id te2 alias tunnel_endpoint2** **(config) # tunnel-endpoint te-id te2 type remote ip-address 1.1.1.201** **(config) # tunnel-endpoint te-id te3 alias tunnel_endpoint3** **(config) # tunnel-endpoint te-id te3 type remote ip-address 1.1.1.202**
5.	Create a port group and specify the list of tunnel endpoints for load balancing.	**(config) # port-group alias pg1 te-list te1,te2,tunnel_endpoint3**

Step	Description	Command
6.	(Optional) Specify weights for each tunnel endpoint in the port group.	**(config) # port-group alias pg1 weight te1 50** **(config) # port-group alias pg1 weight te2 20** **(config) # port-group alias pg1 weight te3 30**
7.	Enable load balancing on the port group.	**(config) # port-group alias pg1 smart-lb enable**
8.	Configure the GigaSMART operation for tunnel encapsulation and assign it to the GigaSMART group. Include the tunnel application, port group, and load balancing metric for stateful load balancing.	**(config) # gsop alias gsop1 tunnel-encap type l2gre pgdst pg1 key 123 session-field ip-any outer lb app tunnel metric round-robin port-list gsport1**
9.	Create a map using the tunnel encapsulation GigaSMART operation.	**(config) # map alias tun_encap** **(config map alias tun_encap) # type regular byRule** **(config map alias tun_encap) # roles replace admin to owner_roles** **(config map alias tun_encap) # use gsop gsop1** **(config map alias tun_encap) # rule add pass ipver 4** **(config map alias tun_encap) # from 1/3/x1** **(config map alias tun_encap) # to 1/3/x2** **(config map alias tun_encap) # exit** **(config) #**
10.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show ip interfaces** **(config) # show tunnel-endpoint** **(config) # show port-group** **(config) # show gsop** **(config) # show map** **(config) # show load-balance port-group stats**

Example 3 – GigaSMART L2GRE Tunnel Encap Stateless LB

Example 3 configures stateless load balancing of tunnel traffic to three tunnel endpoints based on a hash value.

Example 3 differs from Example 2 in the configuration of the GigaSMART operation (gsop).

Step	Description	Command
1.	Configure a tool type of port and a network type of port.	**(config) # port 1/3/x2 type tool** **(config) # port 1/3/x1 type network**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport1 port-list 1/1/e1**
3.	Configure the IP interface with an IP address, subnet mask, default gateway, and MTU setting. Assign it to the GigaSMART group.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/3/x2** **(config ip interface alias test) # ip address 1.1.1.1 /29** **(config ip interface alias test) # gw 1.1.1.100** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsport1** **(config ip interface alias test) # exit**
4.	Configure tunnel endpoints. The alias is optional.	**(config) # tunnel-endpoint te-id te1 alias tunnel_ endpoint1** **(config) # tunnel-endpoint te-id te1 type remote ip-address 1.1.1.200** **(config) # tunnel-endpoint te-id te2 alias tunnel_ endpoint2** **(config) # tunnel-endpoint te-id te2 type remote ip-address 1.1.1.201** **(config) # tunnel-endpoint te-id te3 alias tunnel_ endpoint3** **(config) # tunnel-endpoint te-id te3 type remote ip-address 1.1.1.202**
5.	Create a port group and specify the list of tunnel endpoints for load balancing.	**(config) # port-group alias pg1 te-list te1,te2,tunnel_endpoint3**
6.	Enable load balancing on the port group.	**(config) # port-group alias pg1 smart-lb enable**
7.	Configure the GigaSMART operation for tunnel encapsulation and assign it to the GigaSMART group. Include the tunnel application, port group, and load balancing hashing for stateless load balancing.	**(config) # gsop alias gsop2 tunnel-encap type l2gre pgdst pg1 key 123 lb hash 5-tuple outer port-list gsport1**
8.	Create a map using the tunnel encapsulation GigaSMART operation.	**(config) # map alias tun_encap** **(config map alias tun_encap) # type regular byRule**

Step	Description	Command
		(config map alias tun_encap) # roles replace admin to owner_roles **(config map alias tun_encap) # use gsop gsop2** **(config map alias tun_encap) # rule add pass ipver 4** **(config map alias tun_encap) # from 1/3/x1** **(config map alias tun_encap) # to 1/3/x2** **(config map alias tun_encap) # exit** **(config) #**

Example 4 – GigaSMART L2GRE Tunnel Decapsulation

In this example, an IP interface is configured on the network port. A GigaSMART operation for tunnel decapsulation is configured to decapsulate the filtered packets. A map is configured that uses the L2GRE tunnel decapsulation GigaSMART operation, which receives packets from the remote site over the Internet to the main office using the IP interface with tool port and then forwards packets over the tool port.

Step	Description	Command
1.	Configure a network type of port and a tool type of port.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsport1 port-list 1/1/e1**
3.	Configure the IP interface with an IP address, subnet mask, default gateway, and MTU setting. Assign it to the GigaSMART group. The IP address must match the destination IP address specified at the sending end of the tunnel.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/x3** **(config ip interface alias test) # ip address 2.1.1.1 /29** **(config ip interface alias test) # gw 2.1.1.2** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsport1** **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation for tunnel decapsulation and assign it to the GigaSMART group. The tunnel decapsulation settings include the GRE key that identifies the source of the tunnel.	**(config) # gsop alias tunneldecap tunnel-decap type l2gre key 12314 port-list gsport1**

Step	Description	Command
5.	Create a map using the tunnel decapsulation GigaSMART operation, with packets coming from the Internet through the network port and being sent to the local tool port.	**(config) # map alias tun_decap** **(config map alias tun_decap) # type regular byRule** **(config map alias tun_decap) # use gsop tunneldecap** **(config map alias tun_decap) # rule add pass ipsrc 1.1.1.1 255.255.255.0** **(config map alias tun_decap) # from 1/1/x3** **(config map alias tun_decap) # to 1/1/x4** **(config map alias tun_decap) # exit** **(config) #**
6.	Display the configuration for this example.	**(config) # show ip interfaces** **(config) # show gsop** **(config) # show map**
7.	Display Layer 2 GRE tunnel encapsulation/decapsulation statistics,	config) # show gsop stats alias tunnelencap **(config) # show gsop stats alias tunneldecap**

Example 5 – GigaSMART L2GRE IPv6 Tunnel Encap/Decap

In this example, the encapsulation and decapsulation nodes are configured with IP interfaces using IPv6 addresses.

Step	Description	Command
On the encapsulation node, configure the sending end of the tunnel		
1.	Configure a network type of port and a tool type of port.	**(config) # port 1/3/x7 type network** **(config) # port 1/3/x8 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias grp_en port-list 1/3/e1**
3.	Configure the IP interface with an IPv6 address, prefix length, default gateway, and MTU setting. Assign it to the GigaSMART group.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/3/x7** **(config ip interface alias test) # ipv6 address 2001::2 /64** **(config ip interface alias test) # gw-ipv6 2001::1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add grp_en** **(config ip interface alias test) # exit**

Step	Description	Command
4.	Configure the GigaSMART operation for tunnel encapsulation and assign it to the GigaSMART group.	**(config) # gsop alias gsen tunnel-encap type l2gre ip6dst 2001::3 key 5 port-list grp_en**
5.	Create a map using the tunnel encapsulation GigaSMART operation.	**(config) # map alias map_en** **(config map alias map_en) # type regular byRule** **(config map alias map_en) # use gsop gsen** **(config map alias map_en) # rule add pass ipver 4** **(config map alias map_en) # rule add pass ipver 6** **(config map alias map_en) # from 1/3/x7** **(config map alias map_en) # to 1/3/x8** **(config map alias map_en) # exit** **(config) #**
6.	Display the configuration for this example.	**(config) # show ip interfaces** **(config) # show gsop**
On the decapsulation node, configure the receiving end of the tunnel		
7.	Configure a network type of port and a tool type of port.	**(config) # port 1/4/x24 type network** **(config) # port 1/4/x7 type tool**
8.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias grp_de port-list 1/3/e1**
9.	Configure the IP interface with an IPv6 address, prefix length, default gateway, and MTU setting. Assign it to the GigaSMART group.	**(config) # ip interface alias test1** **(config ip interface alias test1) # attach 1/4/x24** **(config ip interface alias test1) # ipv6 address 2001::3 /64** **(config ip interface alias test1) # gw-ipv6 2001::2** **(config ip interface alias test1) # mtu 9400** **(config ip interface alias test1) # gsgroup add grp_de** **(config ip interface alias test1) # exit**
10.	Configure the GigaSMART operation for tunnel decapsulation and assign it to the GigaSMART group.	**(config) # gsop alias gsde tunnel-decap type l2gre key 5 port-list grp_de**
11.	Create a map using the tunnel decapsulation GigaSMART operation.	**(config) # map alias map_de** **(config map alias map_de) # type regular byRule** **(config map alias map_de) # use gsop gsde** **(config map alias map_de) # rule add pass ipver**

Step	Description	Command
		4 **(config map alias map_de) # rule add pass ipver 6** **(config map alias map_de) # from 1/4/x24** **(config map alias map_de) # to 1/4/x7** **(config map alias map_de) # exit** **(config) #**
12.	Display the configuration for this example.	**(config) # show gsop**
13.	Display IP interface configuration on the encapsulation node **Note**: The show ip interfaces command for an IPv6 tunnel displays the gateway status as Reachable if neighbor discovery is completed with gateway or Not Reachable if neighbor discovery failed. Neighbor discovery is done only on the encapsulation node. On the decapsulation node, the gateway status will be Not Applicable.	**(config) # show ip interface**
14.	Display GigaSMART operation configuration on the decapsulation node	**(config) # show gsop alias gsde**
15.	Display the IP interface statistics	**(config) # show ip interface stats**

IP Fragmentation and Reassembly on L2GRE and GMIP Tunnels

Starting in software version 4.6, L2GRE and GMIP tunnels support IP fragmentation and reassembly of packets. IP fragmentation can occur with encapsulation. Fragmented packets are sent on the tool port at the sending end of the tunnel (for example, at a remote site). IP reassembly occurs with decapsulation. Fragmented packets reaching the network port at the receiving end of the tunnel (for example, at a main office site), are decapsulated and reassembled before being sent to a destination.

Refer to the *"IP Fragmentation and Reassembly on L2GRE and GMIP Tunnels"* section in the *GigaVUE Fabric Management Guide* for more details.

Tunnel Health Checks

Starting in software version 5.3, there are tunnel health checks. The reachability of tunnel destinations is checked and, if the destinations are not reachable, packets will not be sent or will stop being sent.

The tunnel health check on the GigaSMART card defines destinations as follows:

- IP destinations used for sending packets from a single IP interface with tool port to a single IP destination
- tunnel endpoints used for load balancing from a single IP interface with tool port to multiple IP destinations

An SNMP notification can be sent when the status of a tunnel destination or tunnel endpoint changes, either from Up to Down or from Down to Up. Options on the clear command are also added for clearing the destination statistics.

GigaSMART ERSPAN Tunnel Decapsulation

Some Cisco equipment provides the ability to mirror monitored traffic to a remote destination through an ERSPAN tunnel. Using ERSPAN tunnel decapsulation, GigaSMART can act as the receiving end of an ERSPAN tunnel, decapsulating mirrored traffic sent over the Internet from a Cisco switch or router.

ERSPAN Tunnel Header Removal Example

In this example, a tunnel is configured to capture ERSPAN packets, then the ERSPAN header is removed and the packets are forwarded to a tool port.

Step	Description	Command
1.	Configure a network port and tool type of port.	**(config) # port 1/1/g2 type network** **(config) # port 1/1/g1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgp1 port-list 1/3/e1**
3.	Configure the IP interface.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g2** **(config ip interface alias test) # ip address 10.10.10.10 /29** **(config ip interface alias test) # gw 10.10.10.1** **(config ip interface alias test) # mtu 9400**

Step	Description	Command
		(config ip interface alias test) # gsgroup add gsgp1 **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation and assign it to the GigaSMART group. **NOTE:** A flow ID of zero is a wildcard value that matches all flow IDs.	**(config) # gsop alias er1 tunnel-decap type erspan flow-id 0 port-list gsgp1**
5.	Create a map.	**(config) # map alias ermap** **(config map alias ermap) # type regular byRule** **(config map alias ermap) # use gsop er1** **(config map alias ermap) # rule add pass protocol gre** **(config map alias ermap) # from 1/1/g2** **(config map alias ermap) # to 1/1/g1** **(config map alias ermap) # exit** **(config) #**
6.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show ip interfaces** **(config) # show map**

ERSPAN Type III Tunnel Header Removal Example

In this example, a tunnel is configured to capture ERSPAN packets. ERSPAN Type III packets are parsed, the ERSPAN header is removed, and the timestamp is calculated. A timestamp trailer is added before the packets are forwarded to a tool port.

Step	Description	Command
1.	Configure a tool type of port.	**(config) # port 1/1/g1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgp1 port-list 1/3/e1**
3.	Configure the IP interface.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g1** **(config ip interface alias test) # ip address**

Step	Description	Command
		10.10.10.10 /29 **(config ip interface alias test) # gw 10.10.10.1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsgp1** **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation and assign it to the GigaSMART group. **NOTE:** A flow ID of zero is a wildcard value that matches all flow IDs.	**(config) # gsop alias gsop_erspan tunnel-decap type erspan flow-id 0 port-list gsgp1**
5.	Configure a timestamp trailer format.	**(config) # gsparams gsgroup gsgp1 erspan3-timestamp format gs**
6.	Create a map. The map contains a rule to allow marker packets (UDP) to be processed.	**(config) # map alias ermap** **(config map alias ermap) # type regular byRule** **(config map alias ermap) # use gsop gsop_ erspan** **(config map alias ermap) # rule add pass protocol gre** **(config map alias ermap) # rule add pass protocol udp** **(config map alias ermap) # from 1/1/g2** **(config map alias ermap) # to 1/1/g1** **(config map alias ermap) # exit** **(config) #**
7.	View the the ERSPAN III timestamp	**(config) # show gsparams**
8.	View the ERSPAN statistics.	**(config) # show gsop stats alias gsop_erspan**

Refer to the *"ERSPAN Statistics Definitions"* section and to the *"GigaSMART Operations Statistics Definitions"* in the *GigaVUE Fabric Management Guide* for details.

GigaSMART VXLAN Tunnel Decapsulation

Refer to the *"GigaSMART VXLAN Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide* for details.

VXLAN Tunnel Termination Example

Step	Description	Command
1.	Configure a tool type of port.	**(config) # port 1/1/g1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgp1 port-list 1/3/e1**
3.	Configure the IP interface.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g1** **(config ip interface alias test) # ip address 10.10.10.10 /29** **(config ip interface alias test) # gw 10.10.10.1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsgp1** **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias vxlan1 tunnel-decap type vxlan portsrc 200 portdst 4789 vni 200 port-list gsgp1**
5.	Create a map.	**(config) # map alias map** **(config map alias map1) # type regular byRule** **(config map alias map1) # use gsop vxlan1** **(config map alias map1) # rule add pass protocol udp** **(config map alias map1) # from 1/1/g2** **(config map alias map1) # to 1/1/g1** **(config map alias map1) # exit** **(config) #**
6.	View the VXLAN tunnel GSOP.	**(config) # show gsop alias vxlan1**
7.	View the VXLAN tunnel statistics.	**(config) # show gsop stats alias vxlan1**

Refer to the "*Tunnel Decapsulation Statistics Definitions*" section and the"*GigaSMART Operations Statistics Definitions*" in the *GigaVUE Fabric Management Guide* for details.

GigaSMART Custom Tunnel Decapsulation

Use custom tunnel termination to terminate a custom tunnel header that is received at the network IP interface, but is not known to GigaSMART. The destination IP and MAC addresses must match the IP and MAC addresses of the network tunnel.

Refer to the *"GigaSMART Custom Tunnel Decapsulation"* section in the *GigaVUE Fabric Management Guide* for detailed information.

Custom Tunnel Termination Example

Step	Description	Command
1.	Configure a tool type of port.	**(config) # port 1/1/g1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgp1 port-list 1/3/e1**
3.	Configure the IP interface.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g1** **(config ip interface alias test)** **# ip address 10.10.10.10 /29** **(config ip interface alias test) # gw 10.10.10.1** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add gsgp1** **(config ip interface alias test) # exit**
4.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias custom1 tunnel-decap type custom portsrc 200 portdst 200 port-list gsgp1**
5.	Create a map.	**(config) # map alias map2** **(config map alias map2) # type regular byRule** **(config map alias map2) # use gsop custom1** **(config map alias map2) # rule add pass protocol udp** **(config map alias map2) # from 1/1/g2** **(config map alias map2) # to 1/1/g1** **(config map alias map2) # exit** **(config) #**
6.	View the custom tunnel GigaSMART operation.	**(config) # show gsop alias custom1**
7.	View the custom tunnel statistics.	**(config) # show gsop stats alias custom1**

Refer to the *"Tunnel Decapsulation Statistics Definitions"* section and the *"GigaSMART Operations Statistics Definitions"* section in the *GigaVUE Fabric Management Guide* for details.

GigaSMART Header Addition

GigaSMART operations with an **add_header** component can add VLAN tags to packets. This operation is useful in the following situations:

- Differentiating stripped packets from non-stripped packets on common IP ranges (for example,10.x.x.x; 192.168.x.x).

Refer to the *"GigaSMART Header Addition"* section in the *GigaVUE Fabric Management Guide* for detailed information.

GigaSMART De-duplication

GigaSMART de-duplication detects duplicates of the following types:

- IPv4 packets
- IPv6 packets
- non-IP packets (including non-IPv4 and non-IPv6 packets)

Duplicates are packets in which the fields (including the headers and payload) are the same, with the exception of some field such as Time-to-Live (TTL). For example, if two packets are identical except for TTL, they will be counted as duplicates.

GigaSMART De-duplication Example

This example shows the configuration steps for a de-duplication operation in which the GigaSMART application drops duplicate packets.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gs2port1 port-list 2/1/e1**
2.	Configure parameters on the GigaSMART group.	**(config) # gsparams gsgroup gs2port1 dedup-action drop** **(config) # gsparams gsgroup gs2port1 dedup-ip-tos ignore** **(config) # gsparams gsgroup gs2port1 dedup-tcp-seq ignore** **(config) # gsparams gsgroup gs2port1 dedup-vlan ignore** **(config) #** gsparams gsgroup gs2port1 dedup-timer 55000
3.	Configure the GigaSMART operation for de-duplication and assign it to	**(config) # gsop alias testdedup dedup set port-list gs2port1**

Step	Description	Command
	the GigaSMART group. **Note**: For tunneled packets, the outer tunnel/header must be removed before sending them to the de-duplication application to ensure proper packet duplication.	
4.	Create a map.	**(config) # map alias testingdedup** **(config map alias testingdedup) # type regular byRule** **(config map alias testingdedup) # use gsop testdedup** **(config map alias testingdedup) # from 2/2/x4,2/2/x6** **(config map alias testingdedup) # to 2/2/x9** **(config map alias testingdedup) # rule add pass portsrc 443 bidir** **(config map alias testingdedup) # exit** **(config) #**
5.	Display the configuration and statistics for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map** **(config) # show gsop stats** **(config) # show port stats**

GigaSMART Header Stripping

GigaSMART operations with a **strip-header** component can identify and remove headers from tagged packets or headers and trailers from tunneled (encapsulated) packets.

Example – FM6000 Timestamping

The following is an example CLI command to strip packets containing the FM6000 timestamp:

(config) # gsop alias fm6000_replace strip-header fm6000-ts none port-list gsgroup1

The following are example CLI commands to convert packets containing the FM6000 timestamp to UTC and append the UTC timestamp to either the Gigamon trailer or the PRT-H00-X12TS trailer:

(config) # gsop alias fm6000_replace strip-header fm6000-ts gs port-list gsgroup1
(config) # gsop alias fm6000_replace strip-header fm6000-ts x12-ts port-list gsgroup1

The following is an example map using the strip header GigaSMART operation:

```
(config) # map alias fm6000_map
(config map alias fm6000_map) # type regular byRule
(config map alias fm6000_map) # roles replace admin to owner_roles
(config map alias fm6000_map) # use gsop fm6000_replace
(config map alias fm6000_map) # rule add pass ipver 4
(config map alias fm6000_map) # to 1/1/x1
(config map alias fm6000_map) # from 1/1/x2
(config map alias fm6000_map) # exit
(config) #
```

NOTE: There is one-to-one mapping between the GigaSMART operation (gsop) and the map.

If there are multiple devices, each device can be configured with a different timestamp format. To configure this, use a different gsop and a different map for each device. For example, for packets arriving from FM6000 device1, configure a gsop for FM6000 device1 and associate it with map1. For packets arriving from FM6000 device2, configure a gsop for FM6000 device2 and associate it with map2.

All the maps can send all the packets to the same tool port.

Example 1 – Stripping PPPoE Encapsulated Packets

In this example, the PPPoE encapsulated packets are stripped from the packet structure. Figure 15PPPoE Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 15 *PPPoE Encapsulated Packets*

The following is an example CLI command syntax to strip PPPoE encapsulated packets:

```
(config) # gsop alias <alias> strip-header generic anchor-hdr1 eth offset end header-count 1
anchor-hdr2 any port-list gsg
```

Table 14: Components of PPPoE Encapsulated Packets

Component	Description
anchor-hdr1 eth offset end	Starts the Header Stripping operation from the right end of the Ethernet header.
header-count 1	Strips the next header after the Ethernet Header.
anchor-hdr2 any	Updates a valid protocol as the second header in the packet. In this case, any IPv4 or IPv6 protocol can become the second header.

Example 2 – Retaining IPv4 Inner Header from the LISP Header Format

Cisco LISP is used to carry original IP packets to support multi-homing. In this example, the IPv4 outer header, UDP header, and LISP header are stripped from the Cisco LISP header format. The LISP header is considered as an unknown header.

Figure 16Cisco LISP Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 16 *Cisco LISP Encapsulated Packets*

The following is an example CLI command syntax to strip Cisco LISP encapsulated packets:

```
(config) # gsop alias remove_lisp strip-header generic anchor-hdr1 eth offset end header-count 2 custom-len 8 anchor-hdr2 ipv4
```

Table 15: Components of Cisco LISP Encapsulated Packets

Component	Description
anchor-hdr1 eth offset end	Starts the Header Stripping operation from the right end of the Ethernet header.
header-count 2	Strips the next two headers, which are IPv4 Outer Header and UDP from the packet.
custom-len 8	Strips 8 bytes of the unknown packet header. LISP is an unknown header.
anchor-hdr2 ipv4	Updates IPv4 protocol as the second header in the packet.

Example 3 – Stripping Outer MAC Header from the L2 MPLS Encapsulated Frames

The L2 MPLS packet, also known as VPLS, encapsulates Ethernet packets in the MPLS label stack. In this example, the outer Ethernet header and MPLS [PW Label] are stripped from the L2 MPLS encapsulated packets.

Figure 17L2 MPLS Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 17 *L2 MPLS Encapsulated Packets*

The following is an example CLI command syntax to strip the outer MAC header from the L2 MPLS encapsulated packets:

(config) # gsop alias remove_out_mac_vpls strip-header generic anchor-hdr1 none offset start header-count 2 anchor-hdr2 none port-list gsp

Table 16: Components of L2 MPLS Encapsulated Packets

Component	Description
anchor-hdr1 none offset start	Starts the Header Stripping operation from the start of the Ethernet header.
header-count 2	Strips the first and the second header from the packet. The outer Ethernet header and MPLS [PW label] packet header are both removed. As anchor-hdr1 is set to none, the header-count counts the first header for stripping.
anchor-hdr2 none	Signifies that there is no need to specify the second anchor header. In this case, the IPv4 protocol forms the first header of the packet after the stripping operation is complete.

Example 4 – Stripping PW Label Frame from the L2 MPLS Encapsulated Frames

Using the same example as in Example 3 – Stripping Outer MAC Header from the L2 MPLS Encapsulated Frames, another scenario is explained. In this scenario, only the PW Label frame from the MPLS header is removed.

Figure 18L2 MPLS Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 18 *L2 MPLS Encapsulated Packets*

The following is an example CLI command syntax to strip the PW Label frame from the L2 MPLS encapsulated packets:

(config) # gsop alias remove_pwlabel_mpls strip-header generic anchor-hdr1 mpls offset 4 custom-len 4 anchor-hdr2 none port-list gsp

Table 17: Components of L2 MPLS Encapsulated Packets

Component	Description
anchor-hdr1 mpls offset 4	Starts the Header Stripping operation after the first 4 bytes of the MPLS header.
custom-len 4	Strips the custom length of 4 bytes of the PW Label frame.
anchor-hdr2 none	Signifies that it is not necessary to specify the next header as GigaSMART can parse the next header information from the MPLS header.

Example 5 – Retaining Inner Ethernet Frame from the VXLAN Encapsulated Frame

VXLAN encapsulates Ethernet packets in IP using VXLAN header. In this example, the outer Ethernet header, outer IP header, outer UDP header, and VXLAN Header are stripped from the VXLAN encapsulated packets.

Figure 19VXLAN Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

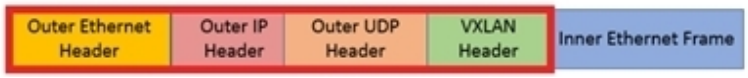


Figure 19 *VXLAN Encapsulated Packets*

The following is an example CLI command syntax to strip outer Ethernet frame from the VXLAN encapsulated packets:

```
(config) # gsop alias remove_outer_mac_vxlan strip-header generic anchor-hdr1 none offset start header-count 4 anchor-hdr2 none port-list gsp
```

Table 18: Components of VXLAN Encapsulated Packets

Component	Description
anchor-hdr1 none offset start	Starts the Header Stripping operation from the start of the Ethernet header.
header-count 4	Strips the next three headers, which is the outer IP header, outer UDP header, and VXLAN header.
anchor-hdr2 none	Signifies that there is no need to specify the second anchor header. In this case, the IPv4 protocol forms the first header of the packet. **Note**: When the first anchor header is set to none, the second anchor header must also be set to none.

Example 6 – Stripping TRILL Header Frames

TRILL encapsulates Ethernet packets in Ethernet frame to provide L2 layer routing in data centers. In this example, consider TRILL frame as an unknown header. This TRILL frame is stripped with the inner Ethernet header from the encapsulated packets. The combined length of TRILL header (6 bytes) and inner Ethernet header (14 bytes) is 20 bytes.

Figure 20TRILL Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 20 *TRILL Encapsulated Packets*

The following is an example CLI command syntax to strip TRILL from the encapsulated packets:

(config) # gsop alias remove_inner_mac_trill strip-header generic anchor-hdr1 eth offset end custom-len 20 anchor-hdr2 ipv4 port-list gsg

Table 19: Componentsof TRILL Encapsulated Packets

Component	Description
anchor-hdr1 eth offset end	Starts the Header Stripping operation from the right end of the outer Ethernet header.
custom-len 20	Strips 20 bytes of unknown header from the packets. In this case, the TRILL frame and the inner Ethernet header is stripped.
anchor-hdr2 ipv4	Updates IPv4 protocol as the second header in the packet.

Example 7 – Stripping Outer Ethernet Header from the Avaya SPB Encapsulated Packets

Avaya SPB (802.1ah) fabric encapsulates Ethernet packets using MAC-In-MAC headers. In this example, the outer Ethernet header and ITAG are removed from the packet structure.

Figure 18L2 MPLS Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

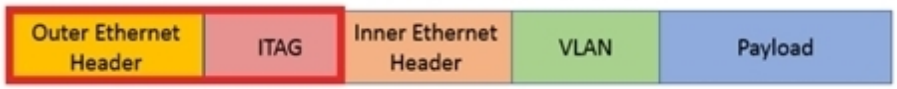


Figure 21 *Avaya SPB Encapsulated Packets*

The following is an example CLI command syntax to strip the outer Ethernet headers from the encapsulated packets:

(config) # gsop alias remove_outer_mac_spb strip-header generic anchor-hdr1 none offset start header-count 2 anchor-hdr2 none port-list gsg

Table 20: Components of Avaya SPB Encapsulated Packets

Component	Description
anchor-hdr1 none offset start	Starts the Header Stripping operation from the left end of the outer Ethernet header.
header-count 2	Strips the outer Ethernet header and ITAG from the packet.
anchor-hdr2 none	Signifies that it is not necessary to specify the next header. The inner Ethernet header becomes the first header after the stripping operation is complete.

Example 8 – Stripping Inner Ethernet Header from the Avaya SPB Encapsulated Packets

Using the same example as in Example 7 – Stripping Outer Ethernet Header from the Avaya SPB Encapsulated Packets, another scenario is explained. In this example, the ITAG, inner Ethernet header, and VLAN are removed from the packet structure.

Figure 18L2 MPLS Encapsulated Packets illustrates a red outline around the frame that needs to be striped.

Figure 22 *Avaya SPB Encapsulated Packets*

The following is an example CLI command syntax to strip the inner Ethernet headers from the encapsulated packets:

```
(config) # gsop alias remove_inner_mac_spb strip-header generic anchor-hdr1 eth end header-count 3 anchor-hdr2 any port-list gsg
```

Table 21: Components of Avaya SPB Encapsulated Packets

Component	Description
anchor-hdr1 eth offset end	Starts the Header Stripping operation from the right end of the outer Ethernet header.
header-count 3	Strips the ITAG, inner Ethernet header, and VLAN from the packet.
anchor-hdr2 any	Indicates that any valid protocol available after the Header Stripping operation can become the next header in the packet.

GigaSMART GTP Correlation

The GigaSMART GTP application correlates traffic based on mobile subscriber IDs in the packet data networks of service providers. It provides a mechanism to filter and forward session traffic for subscribers to tools. GTP correlation assists mobile carriers in debugging and analyzing GTP traffic in their 3G/4G networks.

Refer to the "*GigaSMART GTP Correlation" section* in the *GigaVUE Fabric Management Guide* for detailed information.

Configure GigaSMART GTP Correlation Examples

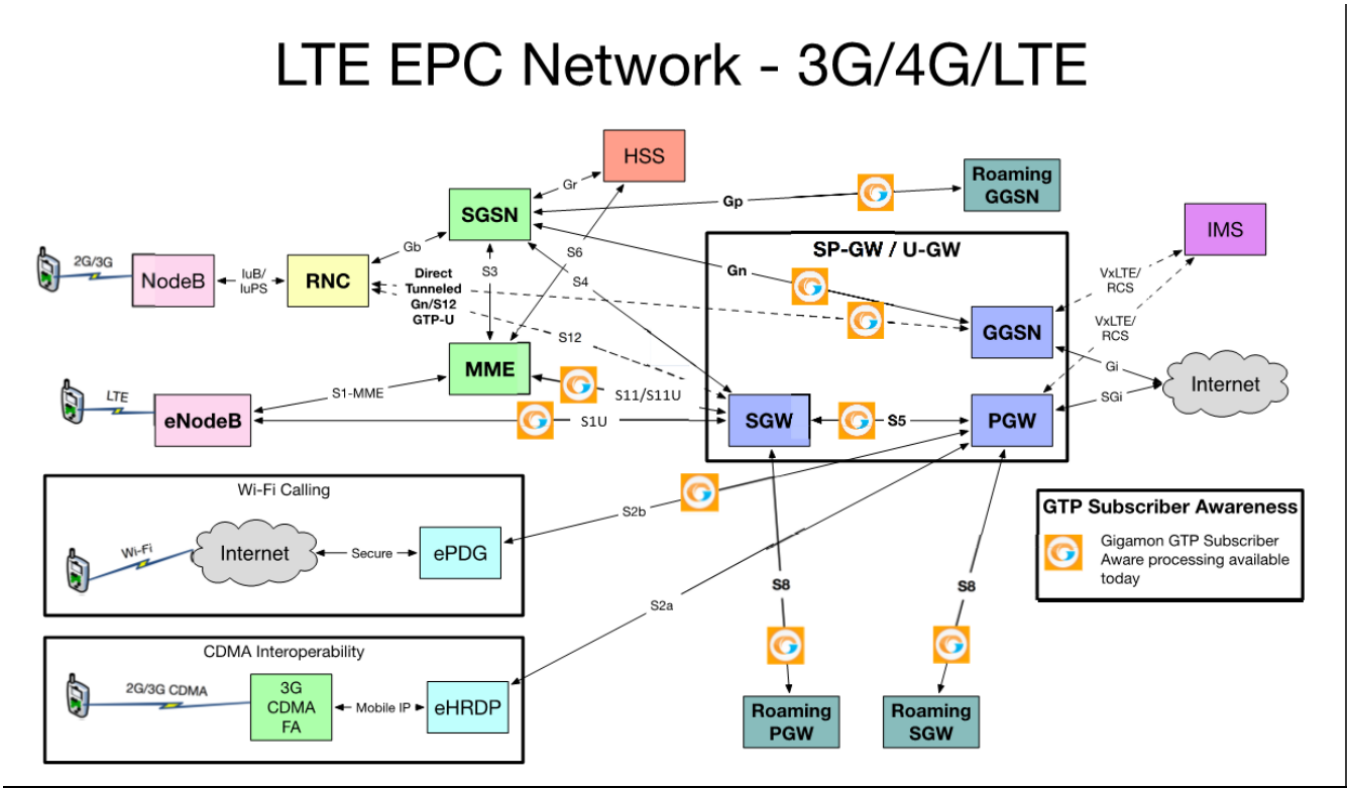
The following sections provide examples of GigaSMART GTP correlation and GigaSMART GTP load balancing.

- Example 1: Identifying High-Value and/or Roaming Subscribers Based on IMSI's
- Example 2: Identifying GTP Versions

- Example 3: Same Subscriber, Filter on Different Versions
- Example 4: Same Subscriber, Filter on Different Interfaces
- Example 5: EPC Filtering
- Example 6: EPC Filtering

Example 1: Identifying High-Value and/or Roaming Subscribers Based on IMSI's

Use GTP correlation to identify high value subscribers based on an IMSI or group of IMSI's. GTP correlation keeps track of the IMSI's that you are interested in monitoring. It correlates them to the corresponding data/user-plane sessions for the subscriber and/or group of subscribers. Filtering on subscriber ID (IMSI) limits the amount of traffic that is sent to monitoring tools.



In Example 1, filter rules are configured to identify and forward all the traffic related to subscribers identified by an IMSI prefix. All traffic specific to the filtered IMSI's 22222222222223*, including GTP-c and GTP-u, is forwarded to a monitoring tool. A shared collector is configured to which traffic not matching the filters is sent.

Step	Description	Command
1.	Configure one network and two tool type of ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool**

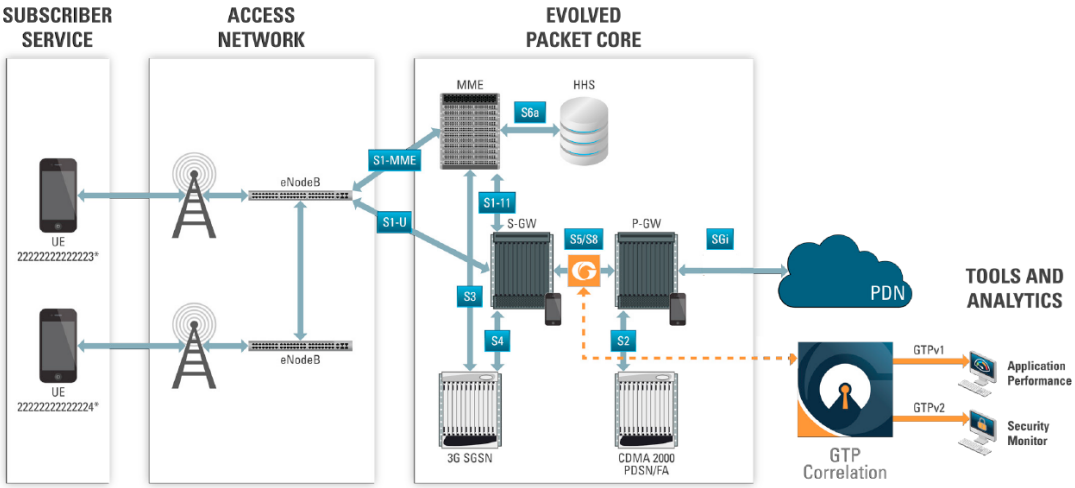
Step	Description	Command
		(config) # port 1/1/x5 type tool
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group to enable GTP correlation.	**(config) # gsop alias gtp_sf flow-ops flow-filtering gtp port-list gsg1**
4.	Configure a virtual port and assign it to the same GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map that directs GTP traffic from physical network port/s to the virtual port you created in the previous step. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # rule add pass portsrc 2123 bidir** **(config map alias to_vp) # rule add pass portsrc 2152 bidir** **(config map alias to_vp) # rule add pass ipfrag all-frag-no-first** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches IMSIs specified by the flow rule, and sends matching traffic to physical tool ports.	**(config) # map alias IMSI-list1** **(config map alias IMSI-list1) # type secondLevel flowFilter** **(config map alias IMSI-list1) # use gsop gtp_sf** **(config map alias IMSI-list1) # to 1/1/x4** **(config map alias IMSI-list1) # from vp1** **(config map alias IMSI-list1) # flowrule add pass gtp imsi 22222222222223*** **(config map alias IMSI-list1) # exit** **(config) #**
7.	Add a shared collector for any unmatched data and send it to the second tool port.	**(config) # map-scollector alias scoll** **(config map-scollector alias scoll) # from vp1** **(config map-scollector alias scoll) # collector 1/1/x5** **(config map-scollector alias scoll) # exit** **(config) #**

Step	Description	Command
8.	Display the configuration for Example 1.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**
9.	Display statistics.	**(config) # show gsgroup flow-ops-report alias gsg1 type flow-filtering any**
10.	Display the session tables for flow-ops-reports.	**(config) # show gsgroup flow-ops-report alias gsg1 type?** **# flow-sampling — Enable flow aware sampling.** **# flow-filtering — Enable flow aware filtering.** **# flow-sip — Fetches a report of SIP/RTP flows for gsgroup.** **# ssl-decryption — Displays Passive SSL decryption.** **# inline-ssl — Choose a inline SSL type.**
11.	Display the GTP correlation statistics associated with the GigaSMART group	**(config) # show gsgroup flow-ops-report alias gsg1 type flow-filtering any**

Refer to the *"Flow Ops Report Statistics Definitions for GTP"* section in the *GigaVUE Fabric Management Guide* for descriptions of these statistics.

Example 2: Identifying GTP Versions

As part of GTP correlation, GigaVUE-OS nodes also provide the flexibility to identify GTPv1 and GTPv2 messages. GTP version information is typically exchanged only as part of the control sessions. By correlating the control and user-plane sessions, GigaVUE-OS nodes can identify, filter, and forward all sessions specific to a GTPv1 or v2 to one or more monitoring/analytic tools.



In Example 2, EMEI traffic is distributed based on GTP versions as follows:

- Filter and forward GTPv1 to a tool port
- Filter and forward GTPv2 to another tool port

Step	Description	Command
1.	Configure one network and two tool type of ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group to enable GTP correlation.	**(config) # gsop alias gtp_sf flow-ops flow-filtering gtp port-list gsg1**
4.	Configure a virtual port and assign it to the same GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map that directs GTP traffic from physical network port/s to the virtual port you created in the previous step. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # rule add pass portsrc 2123 bidir** **(config map alias to_vp) # rule add pass portsrc 2152 bidir** **(config map alias to_vp) # rule add pass ipfrag all-frag-no-first** **(config map alias to_vp) # exit**

Step	Description	Command
		(config) #
6.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches IMEIs specified by the flow rule, and sends matching traffic to a tool port.	**(config) # map alias IMEI-list1** **(config map alias IMEI-list1) # type secondLevel flowFilter** **(config map alias IMEI-list1) # use gsop gtp_sf** **(config map alias IMEI-list1) # to 1/1/x4** **(config map alias IMEI-list1) # from vp1** **(config map alias IMEI-list1) # flowrule add pass gtp imei * version 1** **(config map alias IMEI-list1) # exit** **(config) #**
7.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches IMEIs specified by the flow rule, and sends matching traffic to another tool port.	**(config) # map alias IMEI-list2** **(config map alias IMEI-list2) # type secondLevel flowFilter** **(config map alias IMEI-list2) # use gsop gtp_sf** **(config map alias IMEI-list2) # to 1/1/x1** **(config map alias IMEI-list2) # from vp1** **(config map alias IMEI-list2) # flowrule add pass gtp imei * version 2** **(config map alias IMEI-list2) # exit** **(config) #**
8.	Display the configuration for Example 2.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

Example 3: Same Subscriber, Filter on Different Versions

In this example, traffic from the same subscriber is forwarded to two different load balancing groups based on version. GTP version 1 traffic is sent to one load balancing group and GTP version 2 traffic is sent to another load balancing group.

Step	Description	Command
1.	Configure one network and multiple tool type of ports.	**(config) # port 1/2/g1 type network** **(config) # port 1/2/g5..g9 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e2**

Step	Description	Command
3.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias gtpfilter lb app gtp metric hashing key imsi flow-ops flow-filtering gtp port-list gsgrp1**
4.	Configure a virtual port and assign it to the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create two port groups (one for version 1 traffic and one for version 2 traffic).	**(config) # port-group alias pglbv1 port-list 1/2/g5..g6** **(config) # port-group alias pglbv2 port-list 1/2/g7..g9**
6.	Enable load balancing on the port groups.	**(config) # port-group alias pglbv1 smart-lb enable** **(config) # port-group alias pglbv2 smart-lb enable**
7.	Create an ingress (first level) map.	**(config) # map alias map1_1** **(config map alias map1_1) # type firstLevel byRule** **(config map alias map1_1) # from 1/2/g1** **(config map alias map1_1) # to vp1** **(config map alias map1_1) # rule add pass macdst 00:a0:d1:e1:02:01 0000.0000.0000** **(config map alias map1_1) # exit** **(config) #**
8.	Create a second level map.	**(config) # map alias map2_1** **(config map alias map2_1) # type secondLevel flowFilter** **(config map alias map2_1) # from vp1** **(config map alias map2_1) # use gsop gtpfilter** **(config map alias map2_1) # to pglbv1** **(config map alias map2_1) # flowrule add pass gtp imsi * version 1** **(config map alias map2_1) # exit** **(config) #**
9.	Create another second level map.	**(config) # map alias map2_2** **(config map alias map2_2) # type secondLevel flowFilter** **(config map alias map2_2) # from vp1** **(config map alias map2_2) # use gsop gtpfilter** **(config map alias map2_2) # to pglbv2** **(config map alias map2_2) # flowrule add pass**

Step	Description	Command
		gtp imsi * version 2 **(config map alias map2_2) # exit** **(config) #**

Example 4: Same Subscriber, Filter on Different Interfaces

In this example, traffic from the same subscriber is forwarded to two different load balancing groups based on interface. In this example, VLANs 1601 and 1602 are from S5/S8 interface and VLANs 1611 and 1612 are from S11/S1-U interface. The first level maps split the VLAN traffic to different virtual ports. The second level maps send the traffic to different load balancing groups.

Step	Description	Command
1.	Configure one network and multiple tool type of ports.	**(config) # port 1/2/g1 type network** **(config) # port 1/2/g5..g9 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e2**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias gtpfilter lb app gtp metric hashing key imsi flow-ops flow-filtering gtp port-list gsgrp1**
4.	Configure virtual ports and associate them with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1** **(config) # vport alias vp2 gsgroup gsgrp1**
5.	Create two port groups (one for version 1 traffic and one for version 2 traffic).	**(config) # port-group alias pglbv1 port-list 1/2/g5..g6** **(config) # port-group alias pglbv2 port-list 1/2/g7..g9**
6.	Enable load balancing on the port groups.	**(config) # port-group alias pglbv1 smart-lb enable** **(config) # port-group alias pglbv2 smart-lb enable**
7.	Create a first level map.	**(config) # map alias map1_1** **(config map alias map1_1) # type firstLevel byRule** **(config map alias map1_1) # from 1/2/g1** **(config map alias map1_1) # to vp1** **(config map alias map1_1) # rule add pass vlan 1601..1602**

Step	Description	Command
		(config map alias map1_1) # exit **(config) #**
8.	Create another first level map.	**(config) # map alias map1_2** **(config map alias map1_2) # type firstLevel byRule** **(config map alias map1_2) # from 1/2/g1** **(config map alias map1_2) # to vp2** **(config map alias map1_2) # rule add pass vlan 1611..1612** **(config map alias map1_2) # exit** **(config) #**
9.	Create a second level map.	**(config) # map alias map2_1** **(config map alias map2_1) # type secondLevel flowFilter** **(config map alias map2_1) # from vp1** **(config map alias map2_1) # use gsop gtpfilter** **(config map alias map2_1) # to pglbv1** **(config map alias map2_1) # flowrule add pass gtp imsi *** **(config map alias map2_1) # exit** **(config) #**
10.	Create another second level map.	**(config) # map alias map2_2** **(config map alias map2_2) # type secondLevel flowFilter** **(config map alias map2_2) # from vp2** **(config map alias map2_2) # use gsop gtpfilter** **(config map alias map2_2) # to pglbv2** **(config map alias map2_2) # flowrule add pass gtp imsi *** **(config map alias map2_2) # exit** **(config) #**

Example 5: EPC Filtering

In this example, traffic for all subscribers on interfaces S11/S1-U and Gn/Gp is sent to the same load balancing group. All other traffic is dropped.

Step	Description	Command
1.	Configure one network and two tool	**(config) # port 1/2/g1 type network**

Step	Description	Command
	type of ports.	**(config) # port 1/2/g5..g6 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e2**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias gtpLB lb app gtp metric hashing key imsi flow-ops flow-filtering gtp port-list gsgrp1**
4.	Configure a virtual port and assign it to the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create a port group.	**(config) # port-group alias pglbv1 port-list 1/2/g5..g6**
6.	Enable load balancing on the port group.	**(config) # port-group alias pglbv1 smart-lb enable**
7.	Create an ingress (first level) map. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/2/g1** **(config map alias to_vp) # rule add pass portsrc 2123 bidir** **(config map alias to_vp) # rule add pass portsrc 2152 bidir** **(config map alias to_vp) # exit** **(config) #**
8.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches IMSIs specified by the flow rules, and sends matching traffic to physical tool ports.	**(config) # map alias map2_1** **(config map alias map2_1) # type secondLevel flowFilter** **(config map alias map2_1) # from vp1** **(config map alias map2_1) # use gsop gtpLB** **(config map alias map2_1) # to pglbv1** **(config map alias map2_1) # flowrule add pass gtp imsi * interface Gn** **(config map alias map2_1) # flowrule add pass gtp imsi * interface S11** **(config map alias map2_1) # exit** **(config) #**

Example 6: EPC Filtering

In this example, traffic for all subscribers from all interfaces except S5/S8 is sent to the same load balancing group. Traffic from the S5/S8 interface is dropped.

Step	Description	Command
1.	Configure one network and two tool type of ports.	**(config) # port 1/2/g1 type network** **(config) # port 1/2/g5..g6 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e2**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias gtpLB lb app gtp metric hashing key imsi flow-ops flow-filtering gtp port-list gsgrp1**
4.	Configure a virtual port and assign it to the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create a port group.	**(config) # port-group alias pglbv1 port-list 1/2/g5..g6**
6.	Enable load balancing on the port group.	**(config) # port-group alias pglbv1 smart-lb enable**
7.	Create an ingress (first level) map. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/2/g1** **(config map alias to_vp) # rule add pass portsrc 2123 bidir** **(config map alias to_vp) # rule add pass portsrc 2152 bidir** **(config map alias to_vp) # exit** **(config) #**
8.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches IMSIs specified by the flow rules, and sends matching traffic to physical tool ports.	**(config) # map alias map2_1** **(config map alias map2_1) # type secondLevel flowFilter** **(config map alias map2_1) # from vp1** **(config map alias map2_1) # use gsop gtpLB** **(config map alias map2_1) # to pglbv1** **(config map alias map2_1) # flowrule add drop gtp imsi * interface S5** **(config map alias map2_1) # flowrule add pass gtp imsi ***

Step	Description	Command
		(config map alias map2_1) # exit **(config) #**

GigaSMART GTP Whitelisting and GTP Flow Sampling Examples

GTP whitelisting selects specific subscribers based on IMSI. The whitelist contains up to 500,000 subscriber IMSIs. For subscribers in the whitelist, 100% of their traffic is always sent to a specified tool port.

For example, when a subscriber session comes in, GTP whitelisting checks the IMSI of the subscriber. If the incoming IMSI matches an IMSI in the whitelist, the session is sent to the tool port or load balancing group specified in the whitelist map.

GTP flow sampling samples a configured percentage of GTP sessions. GTP flow sampling uses map rules to select subscribers and then forward a percentage of the packets to tool ports.

Pass rules are defined in flow sampling maps. Each rule contains some combination of IMSI, IMEI, and MSISDN numbers or patterns, Evolved Packet Core (EPC) interface type, GTP version, Access Point Name (APN), or QoS Class Identifier (QCI), as well as a percentage to sample. The flow is sampled to see if it matches a rule. The percentage of the subscriber sessions matching each rule are selected.

Refer to the *"GigaSMART GTP Whitelisting and GTP Flow Sampling"* section in the *GigaVUE Fabric Management Guide* for detailed information.

Refer to the following examples:

- Example 1: GigaSMART GTP Whitelisting
- Example 2: GigaSMART GTP Whitelisting with Multiple Maps
- Example 3: GigaSMART GTP Flow Sampling
- Example 4: GigaSMART GTP Whitelisting, GigaSMART GTP Flow Sampling, and GigaSMART Load Balancing
- Example 5: GigaSMART GTP Flow Sampling with Multiple Maps
- Example 6: GigaSMART GTP Load Balancing in a Cluster
- Example 7: APN for GigaSMART GTP Whitelisting, APN and QCI for GigaSMART GTP Flow Sampling

For details on the CLI commands used in the following examples, refer to the following sections in the CLI reference section:

- apps gtp-whitelist
- gsgroup

- gsop
- gsparams
- map
- port-group
- vport

Example 1: GigaSMART GTP Whitelisting

Example 1 is a GTP whitelisting configuration example. Traffic from network ports go to the three first level maps (GTP-Control, GTP-User, and Fragments-Not-First) and then to the virtual port (vport1). If there is a match to an IMSI in the whitelist (MyIMSIs), it is forwarded to a port.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 10/7/e1**
2.	Create a virtual port.	**(config) # vport alias vport1 gsgroup gsg1**
3.	Create the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs create**
4.	Fetch whitelist files from a specified location to populate the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file1.txt** **(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file2.txt**
5.	Associate the GigaSMART group to the GTP whitelist.	**(config) # gsparams gsgroup gsg1 gtp-whitelist add MyIMSIs**
6.	Configure the GigaSMART operation for GTP whitelisting.	**(config) #** gsop alias gtp-whitelist flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg1
7.	Configure three first level maps. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias GTP-Control** **(config map alias GTP-Control) # type firstLevel byRule** **(config map alias GTP-Control) # roles replace admin to owner_roles** **(config map alias GTP-Control) # rule add pass portdst 2123 bidir** **(config map alias GTP-Control) # to vport1** **(config map alias GTP-Control) # from 8/1/x40,8/1/x6** **(config map alias GTP-Control) # exit** **(config) #** **(config) # map alias GTP-User** **(config map alias GTP-User) # type firstLevel byRule**

Step	Description	Command
		(config map alias GTP-User) # roles replace admin to owner_roles **(config map alias GTP-User) # rule add pass portdst 2152 bidir** **(config map alias GTP-User) # to vport1** **(config map alias GTP-User) # from 8/1/x40,8/1/x6** **(config map alias GTP-User) # exit** **(config) #** **(config) # map alias Fragments-Not-First** **(config map alias Fragments-Not-First) # type firstLevel byRule** **(config map alias Fragments-Not-First) # roles replace admin to owner_roles** **(config map alias Fragments-Not-First) # rule add pass ipfrag all-frag-no-first** **(config map alias Fragments-Not-First) # to vport1** **(config map alias Fragments-Not-First) # from 8/1/x40,8/1/x6** **(config map alias Fragments-Not-First) # exit** **(config) #**
8.	Configure a second level map for GTP whitelisting, the whitelist map. If there is a match to an IMSI in the whitelist (MyIMSIs), it is forwarded to a port.	**(config) # map alias GTP-Whitelist** **(config map alias GTP-Whitelist) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist) # roles replace admin to owner_roles** **(config map alias GTP-Whitelist) # use gsop gtp-whitelist** **(config map alias GTP-Whitelist) # to 10/5/x17** **(config map alias GTP-Whitelist) # from vport1** **(config map alias GTP-Whitelist) # exit** **(config) #**

Example 2: GigaSMART GTP Whitelisting with Multiple Maps

Example 2 is a GTP whitelisting configuration example that includes multiple GTP whitelisting maps, which provide a more granular selection of tool ports.

Traffic from network ports go to the three first level maps (GTP-Control, GTP-User, and Fragments-Not-First) and then to the virtual port (vport1). Two whitelist maps are configured. The first map specifies a rule for version 1 traffic. The second map specifies a rule for version 2 traffic.

The following table provide step-by-step configuration for GTP flow sampling using GigaSMART maps.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 10/7/e1**
2.	Create a virtual port.	**(config) # vport alias vport1 gsgroup gsg1**
3.	Create the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs create**
4.	Fetch whitelist files from a specified location to populate the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file1.txt** **(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file2.txt**
5.	Associate the GigaSMART group to the GTP whitelist.	**(config) # gsparams gsgroup gsg1 gtp-whitelist add MyIMSIs**
6.	Configure the GigaSMART operation for GTP whitelisting.	**(config) #** gsop alias gtp-whitelist flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg1
7.	Configure three first level maps. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias GTP-Control** **(config map alias GTP-Control) # type firstLevel byRule** **(config map alias GTP-Control) # roles replace admin to owner_roles** **(config map alias GTP-Control) # rule add pass portdst 2123 bidir** **(config map alias GTP-Control) # to vport1** **(config map alias GTP-Control) # from 8/1/x40,8/1/x6** **(config map alias GTP-Control) # exit** **(config) #** **(config) # map alias GTP-User** **(config map alias GTP-User) # type firstLevel byRule** **(config map alias GTP-User) # roles replace admin to owner_roles** **(config map alias GTP-User) # rule add pass portdst 2152 bidir** **(config map alias GTP-User) # to vport1** **(config map alias GTP-User) # from 8/1/x40,8/1/x6** **(config map alias GTP-User) # exit** **(config) #** **(config) # map alias Fragments-Not-First** **(config map alias Fragments-Not-First) # type**

Step	Description	Command
		firstLevel byRule **(config map alias Fragments-Not-First) # roles replace admin to owner_roles** **(config map alias Fragments-Not-First) # rule add pass ipfrag all-frag-no-first** **(config map alias Fragments-Not-First) # to vport1** **(config map alias Fragments-Not-First) # from 8/1/x40,8/1/x6** **(config map alias Fragments-Not-First) # exit** **(config) #**
8.	Configure one second level map for GTP whitelisting, the first whitelist map. If there is a match to version 1 and if the IMSI is present in the whitelist (MyIMSIs), it is forwarded to the specified port.	**(config) # map alias GTP-Whitelist_v1** **(config map alias GTP-Whitelist_v1) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist_v1) # roles replace admin to owner_roles** **(config map alias GTP-Whitelist_v1) # use gsop gtp-whitelist** **(config map alias GTP-Whitelist_v1) # to 1/2/x23** **(config map alias GTP-Whitelist_v1) # from vport1** **(config map alias GTP-Whitelist_v1) # whitelist add gtp version 1** **(config map alias GTP-Whitelist_v1) # exit** **(config) #**
9.	Configure another second level map for GTP whitelisting, the second whitelist map. If there is a match to version 2 and if the IMSI is present in the whitelist (MyIMSIs), it is forwarded to the specified port.	**(config) # map alias GTP-Whitelist_v2** **(config map alias GTP-Whitelist_v2) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist_v2) # roles replace admin to owner_roles** **(config map alias GTP-Whitelist_v2) # use gsop gtp-whitelist** **(config map alias GTP-Whitelist_v2) # to 1/2/x24** **(config map alias GTP-Whitelist_v2) # from vport1** **(config map alias GTP-Whitelist_v2) # whitelist add gtp version 2** **(config map alias GTP-Whitelist_v2) # exit** **(config) #**

Example 3: GigaSMART GTP Flow Sampling

Example 3 is a GTP flow sampling configuration example. Traffic from network ports go to the three first level maps (GTP-Control, GTP-User, and Fragments-Not-First) and then to the virtual port (vport1). The traffic flow is sampled based on the rules in one flow sampling map

(GTP-Sample-01). The flow sampling rules specify IMSI, IMEI, and MSISDN numbers, as well as the percentage to sample. Packets are then accepted or rejected. Accepted packets are forwarded to a port. Rejected packets are dropped. Packets that do not match a rule will be passed to subsequent maps.

The following table provides step-by-step configuration for GTP flow sampling using GigaSMART maps.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 10/7/e1**
2.	Create a virtual port.	**(config) # vport alias vport1 gsgroup gsg1**
3.	Configure three first level maps. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias GTP-Control** **(config map alias GTP-Control) # type firstLevel byRule** **(config map alias GTP-Control) # roles replace admin to owner_roles** **(config map alias GTP-Control) # rule add pass portdst 2123 bidir** **(config map alias GTP-Control) # to vport1** **(config map alias GTP-Control) # from 8/1/x40,8/1/x6** **(config map alias GTP-Control) # exit** **(config) #** **(config) # map alias GTP-User** **(config map alias GTP-User) # type firstLevel byRule** **(config map alias GTP-User) # roles replace admin to owner_roles** **(config map alias GTP-User) # rule add pass portdst 2152 bidir** **(config map alias GTP-User) # to vport1** **(config map alias GTP-User) # from 8/1/x40,8/1/x6** **(config map alias GTP-User) # exit** **(config) #** **(config) # map alias Fragments-Not-First** **(config map alias Fragments-Not-First) # type firstLevel byRule** **(config map alias Fragments-Not-First) # roles replace admin to owner_roles** **(config map alias Fragments-Not-First) # rule add pass ipfrag all-frag-no-first**

Step	Description	Command
		(config map alias Fragments-Not-First) # to vport1 **(config map alias Fragments-Not-First) # from 8/1/x40,8/1/x6** **(config map alias Fragments-Not-First) # exit** **(config) #**
4.	Configure the GigaSMART operation for GTP flow sampling.	**(config) #** gsop alias gtp-flowsample flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list gsg1
5.	Configure a second level map for GTP flow sampling, the flow sampling map. The traffic flow is sampled based on the rules in this map.	**(config) # map alias GTP-Sample-01** **(config map alias GTP-Sample-01) # type secondLevel flowSample** **(config map alias GTP-Sample-01) # roles replace admin to owner_roles** **(config map alias GTP-Sample-01) # use gsop gtp-flowsample** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* imei 01416800* percentage 50** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 46* percentage 80** **(config map alias GTP-Sample-01) # flowsample add gtp msisdn 1509* percentage 25** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* imei 01400* percentage 15** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* msisdn 1909* percentage 20** **(config map alias GTP-Sample-01) # to 10/5/x18** **(config map alias GTP-Sample-01) # from vport1** **(config map alias GTP-Sample-01) # exit** **(config) #**

Example 4: GigaSMART GTP Whitelisting, GigaSMART GTP Flow Sampling, and GigaSMART Load Balancing

Example 4 combines the GTP whitelisting configuration from Example 1 with the GTP flow sampling configuration from Example 3, and adds GigaSMART load balancing.

In Example 4, traffic from network ports go to the three first level maps (GTP-Control, GTP-User, and Fragments-Not-First) and then to the virtual port (vport1). If there is a match to an IMSI in the whitelist (MyIMSIs), it is forwarded to the port group (PG-Whitelist) for load balancing.

NOTE: In Example 4, the tool ports in the port group are on the same node as the GigaSMART group and GigaSMART operation.

If there is not a match to an IMSI in the whitelist, the traffic flow is sampled based on the rules in the flow sampling map (GTP-Sample-01). The flow sampling rules specify IMSI, IMEI, and MSISDN numbers, as well as the percentage to sample. Packets are then accepted or rejected. Accepted packets are forwarded to the port group (PG-Sample) for load balancing. Rejected packets are dropped. Packets that do not match a rule will be passed to subsequent maps.

The following table provides step-by-step configuration for GTP whitelisting, flow sampling, and load balancing using GigaSMART.

Step	Description	Command
1.	Create port groups and specify the tool ports for load balancing.	**(config) # port-group alias PG-Whitelist port-list 10/5/x17..x18** **(config) # port-group alias PG-Sample port-list 10/5/x19..x20**
2.	Enable load balancing on the port groups.	**(config) # port-group alias PG-Whitelist smart-lb enable** **(config) # port-group alias PG-Sample smart-lb enable**
3.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 10/7/e1**
4.	Create a virtual port.	**(config) # vport alias vport1 gsgroup gsg1**
5.	Configure three first level maps. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias GTP-Control** **(config map alias GTP-Control) # type firstLevel byRule** **(config map alias GTP-Control) # roles replace admin to owner_roles** **(config map alias GTP-Control) # rule add pass portdst 2123 bidir** **(config map alias GTP-Control) # to vport1** **(config map alias GTP-Control) # from 8/1/x40,8/1/x6** **(config map alias GTP-Control) # exit** **(config) #** **(config) # map alias GTP-User** **(config map alias GTP-User) # type firstLevel byRule** **(config map alias GTP-User) # roles replace admin to owner_roles** **(config map alias GTP-User) # rule add pass portdst 2152 bidir** **(config map alias GTP-User) # to vport1** **(config map alias GTP-User) # from 8/1/x40,8/1/x6**

Step	Description	Command
		(config map alias GTP-User) # exit **(config) #** **(config) # map alias Fragments-Not-First** **(config map alias Fragments-Not-First) # type firstLevel byRule** **(config map alias Fragments-Not-First) # roles replace admin to owner_roles** **(config map alias Fragments-Not-First) # rule add pass ipfrag all-frag-no-first** **(config map alias Fragments-Not-First) # to vport1** **(config map alias Fragments-Not-First) # from 8/1/x40,8/1/x6** **(config map alias Fragments-Not-First) # exit** **(config) #**
6.	Create the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs create**
7.	Fetch whitelist files from a specified location to populate the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file1.txt** **(config) # apps gtp-whitelist alias MyIMSIs fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file2.txt**
8.	(Optional) Add a single IMSI to the GTP whitelist.	**(config) # apps gtp-whitelist alias MyIMSIs add imsi 318260109318283**
9.	Associate the GigaSMART group to the GTP whitelist.	**(config) # gsparams gsgroup gsg1 gtp-whitelist add MyIMSIs**
10.	Configure the GigaSMART operation for GTP whitelisting.	**(config) #** gsop alias gtp-whitelist flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg1
11.	Configure a second level map for GTP whitelisting, the whitelist map. If there is a match to an IMSI in the whitelist (MyIMSIs), it is forwarded to a load balancing port group.	**(config) # map alias GTP-Whitelist** **(config map alias GTP-Whitelist) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist) # roles replace admin to owner_roles** **(config map alias GTP-Whitelist) # use gsop gtp-whitelist** **(config map alias GTP-Whitelist) # to PG-Whitelist** **(config map alias GTP-Whitelist) # from vport1** **(config map alias GTP-Whitelist) # exit** **(config) #**
12.	Configure the GigaSMART operation for GTP flow	**(config) #** gsop alias gtp-flowsample flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list gsg1

Step	Description	Command
	sampling.	
13.	Configure a second level map for GTP flow sampling, the flow sampling map. If there is not a match to an IMSI in the whitelist, the traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to a load balancing port group.	**(config) # map alias GTP-Sample-01** **(config map alias GTP-Sample-01) # type secondLevel flowSample** **(config map alias GTP-Sample-01) # roles replace admin to owner_roles** **(config map alias GTP-Sample-01) # use gsop gtp-flowsample** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* imei 01416800* percentage 50** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 46* percentage 80** **(config map alias GTP-Sample-01) # flowsample add gtp msisdn 1509* percentage 25** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* imei 01400* percentage 15** **(config map alias GTP-Sample-01) # flowsample add gtp imsi 31* msisdn 1909* percentage 20** **(config map alias GTP-Sample-01) # to PG-Sample** **(config map alias GTP-Sample-01) # from vport1** **(config map alias GTP-Sample-01) # exit** **(config) #**
14.	Display the configuration for this example.	**(config) # show port-group** **(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show gsparams** **(config) # show map** **(config) # show gsgroup flow-whitelist** **(config) # show map stats all** **(config) # show apps gtp-whitelist**
15.	Display the GTP whitelist entry count	**(config) # show apps gtp-whitelist alias MyIMSIs count**

NOTE: IP-CAN-Bearer—IP Connectivity Access Network (CAN) Bearer, refers to bearers in 3G/4G. With the introduction of APN filtering, GTP correlation started to be based on bearers and not subscribers. Specific bearers of subscribers are counted as bearers matched under the IP-CAN-Bearer heading.

Example 5: GigaSMART GTP Flow Sampling with Multiple Maps

Example 5 includes multiple GTP flow sampling maps, which provide a more granular selection of tool ports for flow sampling.

In Example 5, traffic from network ports go to the three first level maps (GTP-Control, GTP-User, and Fragments-Not-First) and then to the virtual port (vport1). If there is a match to an IMSI in the whitelist (VoLTE_IMM), it is forwarded to the port group (PG-Whitelist-1) for load balancing.

NOTE: In Example 5, the tool ports in the port group are on the same node as the GigaSMART group and GigaSMART operation.

If there is not a match to an IMSI in the whitelist, the traffic flow is sampled based on the rules in four flow sampling maps (GTP-Sample-1 to GTP-Sample-4).

The flow sampling rules in each map specify IMSI, IMEI, and MSISDN numbers, as well as the percentage to sample. Packets are then accepted or rejected. Accepted packets are forwarded to the port group (PG-Sample-1 to PG-Sample-4) for load balancing. Rejected packets are dropped. Packets that do not match a rule will be passed to subsequent maps, in this example, to a shared collector.

The following table provides step-by-step configuration for multi-map GTP flow sampling and load balancing using GigaSMART.

Step	Description	Command
1.	Create port groups and specify the tool ports for load balancing.	**(config) # port-group alias PG-Sample-1 port-list 10/5/x17..x20** **(config) # port-group alias PG-Sample-2 port-list 10/5/x21..x22** **(config) # port-group alias PG-Sample-3 port-list 10/4/x5..x6** **(config) # port-group alias PG-Sample-4 port-list 10/4/x7..x8** **(config) # port-group alias PG-Whitelist-1 port-list 10/5/x23..x24**
2.	Enable load balancing on the port groups.	**(config) # port-group alias PG-Sample-1 smart-lb enable** **(config) # port-group alias PG-Sample-2 smart-lb enable** **(config) # port-group alias PG-Sample-3 smart-lb enable** **(config) # port-group alias PG-Sample-4 smart-lb enable**

Step	Description	Command
		(config) # port-group alias PG-Whitelist-1 smart-lb enable
3.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 10/7/e1**
4.	Create a virtual port.	**(config) # vport alias vport1 gsgroup gsg1**
5.	Configure three first level maps. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias GTP-Control** **(config map alias GTP-Control) # type firstLevel byRule** **(config map alias GTP-Control) # roles replace admin to owner_roles** **(config map alias GTP-Control) # rule add pass portdst 2123 bidir** **(config map alias GTP-Control) # to vport1** **(config map alias GTP-Control) # from 10/1/x5,10/3/x1,10/6/q1** **(config map alias GTP-Control) # exit** **(config) #** **(config) # map alias GTP-User** **(config map alias GTP-User) # type firstLevel byRule** **(config map alias GTP-User) # roles replace admin to owner_roles** **(config map alias GTP-User) # rule add pass portdst 2152 bidir** **(config map alias GTP-User) # to vport1** **(config map alias GTP-User) # from 10/1/x5,10/3/x1,10/6/q1** **(config map alias GTP-User) # exit** **(config) #** **(config) # map alias Fragments-Not-First** **(config map alias Fragments-Not-First) # type firstLevel byRule** **(config map alias Fragments-Not-First) # roles replace admin to owner_roles** **(config map alias Fragments-Not-First) # rule add pass ipfrag all-frag-no-first** **(config map alias Fragments-Not-First) # to vport1** **(config map alias Fragments-Not-First) # from 10/1/x5,10/3/x1,10/6/q1** **(config map alias Fragments-Not-First) # exit**

Step	Description	Command
		(config) #
6.	Create the GTP whitelist.	**(config) # apps gtp-whitelist alias VoLTE_1MM create**
7.	Fetch whitelist files from a specified location to populate the GTP whitelist.	**(config) # apps gtp-whitelist alias VoLTE_1MM fetch add http://10.1.1.100/tftpboot/myfiles/IMSI_file1.txt** **(config) # apps gtp-whitelist alias VoLTE_1MM fetch add http://10.1.1.100/tftpboot/myfiles/IMSI_file2.txt**
8.	(Optional) Add a single IMSI to the GTP whitelist.	**(config) # apps gtp-whitelist alias VoLTE_1MM add imsi 318260109318283**
9.	Associate the GigaSMART group to the GTP whitelist.	**(config) # gsparams gsgroup gsg1 gtp-whitelist add VoLTE_1MM**
10.	Configure the GigaSMART operation for GTP whitelisting.	**(config) #** gsop alias gtp-whitelist-1 flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg1
11.	Configure a second level map for GTP whitelisting, the whitelist map. If there is a match to an IMSI in the whitelist (VoLTE_1MM), it is forwarded to a load balancing port group.	**(config) # map alias GTP-Whitelist** **(config map alias GTP-Whitelist) # type secondLevel flowWhitelist** **(config map alias GTP-Whitelist) # roles replace admin to owner_roles** **(config map alias GTP-Whitelist) # use gsop gtp-whitelist-1** **(config map alias GTP-Whitelist) # to PG-Whitelist-1** **(config map alias GTP-Whitelist) # from vport1** **(config map alias GTP-Whitelist) # exit** **(config) #**
12.	Configure the GigaSMART operation for GTP flow sampling.	**(config) #** gsop alias gtp-flowsample-1 flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list gsg1
13.	Configure a second level map for GTP flow sampling, the first flow sampling map. This map has 12 rules. Traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to a load balancing port group.	**(config) # map alias GTP-Sample-1** **(config map alias GTP-Sample-1) # type secondLevel flowSample** **(config map alias GTP-Sample-1) # roles replace admin to owner_roles** **(config map alias GTP-Sample-1) # use gsop gtp-flowsample-1** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 3182609833* imei 35609506* percentage 75** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 3182609834* imei 3560950* percentage 10** **(config map alias GTP-Sample-1) # flowsample add**

Step	Description	Command
		gtp imsi 31826098350* imei 356095* percentage 20 **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098351* imei 35609* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098352* imei 3560* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098353* imei 356* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098354* imei 35* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098355* imei 3* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098356* imei 356095* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098357* imei 3560* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098358* imei 35* percentage 20** **(config map alias GTP-Sample-1) # flowsample add gtp imsi 31826098359* imei 356095* percentage 20** **(config map alias GTP-Sample-1) # to PG-Sample-1** **(config map alias GTP-Sample-1) # from vport1** **(config map alias GTP-Sample-1) # exit** **(config) #**
14.	Configure a second level map for GTP flow sampling, the second flow sampling map. This map has 12 rules. Traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to a load balancing port group.	**(config) # map alias GTP-Sample-2** **(config map alias GTP-Sample-2) # type secondLevel flowSample** **(config map alias GTP-Sample-2) # roles replace admin to owner_roles** **(config map alias GTP-Sample-2) # use gsop gtp-flowsample-1** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 3182609836* imei 35609506* percentage 30** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 3182609837* imei 356095062* percentage 5** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098380* imei 356095062* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098381* imei 35609506* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098382* imei 3560950* percentage 50**

Step	Description	Command
		(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098383* imei 356095* percentage 50 **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098384* imei 35* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098385* imei 356* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098386* imei 3560* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098387* imei 35609* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098388* imei 356095* percentage 50** **(config map alias GTP-Sample-2) # flowsample add gtp imsi 31826098389* imei 3560950* percentage 50** **(config map alias GTP-Sample-2) # to PG-Sample-2** **(config map alias GTP-Sample-2) # from vport1** **(config map alias GTP-Sample-2) # exit** **(config) #**
15.	Configure a second level map for GTP flow sampling, the third flow sampling map. This map has 5 rules. Traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to a load balancing port group.	**(config) # map alias GTP-Sample-3** **(config map alias GTP-Sample-3) # type secondLevel flowSample** **(config map alias GTP-Sample-3) # roles replace admin to owner_roles** **(config map alias GTP-Sample-3) # use gsop gtp-flowsample-1** **(config map alias GTP-Sample-3) # flowsample add gtp imsi 31826098390* imei 35609506* percentage 10** **(config map alias GTP-Sample-3) # flowsample add gtp imsi 31826098391* imei 35609506* percentage 10** **(config map alias GTP-Sample-3) # flowsample add gtp imsi 31826098392* imei 35609506* percentage 10** **(config map alias GTP-Sample-3) # flowsample add gtp imsi 31826098393* imei 35609506* percentage 10** **(config map alias GTP-Sample-3) # flowsample add gtp imsi 31826098394* imei 35609506* percentage 10** **(config map alias GTP-Sample-3) # to PG-Sample-3** **(config map alias GTP-Sample-3) # from vport1** **(config map alias GTP-Sample-3) # exit**

Step	Description	Command
		(config) #
16.	Configure a second level map for GTP flow sampling, the fourth flow sampling map. This map has one rule. Traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to a load balancing port group.	**(config) # map alias GTP-Sample-4** **(config map alias GTP-Sample-4) # type secondLevel flowSample** **(config map alias GTP-Sample-4) # roles replace admin to owner_roles** **(config map alias GTP-Sample-4) # use gsop gtp-flowsample-1** **(config map alias GTP-Sample-4) # flowsample add gtp imsi 31826098429* imei 35609506* percentage 10** **(config map alias GTP-Sample-4) # to PG-Sample-4** **(config map alias GTP-Sample-4) # from vport1** **(config map alias GTP-Sample-4) # exit** **(config) #**
17.	Configure a collector map for any packets that do not match other rules.	**(config) # map alias GTP-Collector** **(config map alias GTP-Collector) # roles replace admin to owner_roles** **(config map alias GTP-Collector) # from vport1** **(config map alias GTP-Collector) # collector gtp-collector** **(config map alias GTP-Collector) # exit** **(config) #**
18.	Display the configuration for this example.	**(config) # show port-group** **(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show gsparams** **(config) # show map**

Example 6: GigaSMART GTP Load Balancing in a Cluster

Example 6 includes GTP load balancing in a cluster. The tool ports in the port groups must be on the same node, but the GigaSMART group and GigaSMART operation can be on a different node.

GTP load balancing in a cluster is supported for GTP flow filtering and GTP flow sampling.

In Example 6, two nodes are in a cluster, connected through a stack link. The port groups are specified in the **to** parameter of second level maps.

For information on GigaSMART load balancing, refer to GigaSMART Load Balancing.

NOTE: When the **show load-balance port-group stats all** command is executed from the leader, statistics for all the attached load balanced port groups are displayed. When the command is executed from another node in the cluster (standby or normal), only the statistics for the load balanced port group in the map attached to the GigaSMART operation on that node are displayed. The results displayed for the **show load-balance port-group stats alias** <alias> command are similar.

The following table provides step-by-step configuration for GTP load balancing in a clustered environment using GigaSMART.

Step	Description	Command
1.	Configure ports on two nodes as follows: ▪ network ports on node 1. These will be used in first level maps for GTP flow filtering and flow sampling. ▪ tool ports on node 2. These will be used in port groups and GigaStreams. ▪ stack ports on node 1 and node 2. These will be used in GigaStreams. Then administratively enable the ports.	**(config) # port 1/1/q1 type network** **(config) # port 1/1/x6 type network** **(config) # port 1/1/x8 type network** **(config) # port 1/1/x10 type network** **(config) # port 1/1/x12 type network** **(config) # port 2/6/x1..x4 type tool** **(config) # port 2/7/x1..x6 type tool** **(config) # port 2/8/x1..x8 type tool** **(config) # port 1/2/q1..q8 type stack** **(config) # port 2/1/q1..q8 type stack** **(config) # port 1/1/q1 params admin enable** **(config) # port 1/1/x6 params admin enable** **(config) # port 1/1/x8 params admin enable** **(config) # port 1/1/x10 params admin enable** **(config) # port 1/1/x12 params admin enable** **(config) # port 2/6/x1..x4,2/7/x1..x6,2/8/x1..x8 params admin enable** **(config) # port 1/2/q1..q8,2/1/q1..q8 params admin enable**
2.	Configure GigaStreams as follows: ▪ The first two GigaStreams will be used in shared collectors, one for flow filtering and one for flow sampling. ▪ The next two GigaStreams will be used in the stack link	**(config) # gigastream alias GTP-Collector-Filter port-list 2/8/x5..x6 params hash advanced** **(config) # gigastream alias GTP-Collector-Sample port-list 2/7/x5..x6 params hash advanced** **(config) # gigastream alias gstrm_stck_1_2_q1q8 port-list 1/2/q1..q8 params hash advanced** **(config) # gigastream alias gstrm_stck_2_1_q1q8 port-list 2/1/q1..q8 params hash advanced**

Step	Description	Command
	between the two nodes.	
3.	Create three port groups and specify four tool ports each, for load balancing. Also, enable load balancing on each port group. The port groups are as follows: ▪ The first port group is the destination for a second level map for flow filtering version 1. ▪ The second port group is the destination for a second level map for flow filtering version 2. ▪ The third port group is the destination for a second level map for flow sampling. **Note**: The tool ports in the port groups must be on the same node.	**(config) # port-group alias PG-Filter-Version1** **(config port-group alias PG-Filter-Version1) # port-list 2/8/x1..x4** **(config port-group alias PG-Filter-Version1) # smart-lb enable** **(config port-group alias PG-Filter-Version1) # exit** **(config) # port-group alias PG-Filter-Version2** **(config port-group alias PG-Filter-Version2) # port-list 2/6/x1..x4** **(config port-group alias PG-Filter-Version2) # smart-lb enable** **(config port-group alias PG-Filter-Version2) # exit** **(config) # port-group alias PG-Sample** **(config port-group alias PG-Sample) # port-list 2/7/x1..x4** **(config port-group alias PG-Sample) # smart-lb enable** **(config port-group alias PG-Sample) # exit**
4.	Configure the stack link between the nodes.	**(config) # stack-link alias stck_lnk_bn_1and3 between gigastreams gstrm_stck_1_2_q1q8 and gstrm_stck_2_1_q1q8**
5.	Configure two GigaSMART groups, one for GTP flow filtering and one for GTP flow sampling. **Note**: The GigaSMART groups are on a different node than the port groups.	**(config) # gsgroup alias GSG-Filter port-list 1/3/e1** **(config) # gsgroup alias GSG-Sample port-list 1/4/e1**
6.	Configure a flow filtering GigaSMART operation, specify load balancing, and assign the GigaSMART operation to the GigaSMART group for flow filtering. Configure a flow sampling GigaSMART operation, specify load balancing, and assign the GigaSMART operation to the GigaSMART group for flow sampling.	**(config) # gsop alias GSOP-Filter flow-ops flow-filtering gtp lb app gtp metric hashing key imsi port-list GSG-Filter** **(config) # gsop alias GSOP-Sample flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list GSG-Sample**

Step	Description	Command
	Note: The GigaSMART operations are on a different node than the port groups.	
7.	Configure virtual ports and assign them to the same GigaSMART groups.	**(config) # vport alias VP-Filter gsgroup GSG-Filter** **(config) # vport alias VP-Sample gsgroup GSG-Sample**
8.	Create a first level map that directs GTP traffic from the network ports to the virtual port for flow filtering. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias Map-Lev1-GTP-Filter** **(config map alias Map-Lev1-GTP-Filter) # type firstLevel byRule** **(config map alias Map-Lev1-GTP-Filter) # roles replace admin to owner_roles** **(config map alias Map-Lev1-GTP-Filter) # rule add pass portdst 2123 bidir** **(config map alias Map-Lev1-GTP-Filter) # rule add pass portdst 2152 bidir** **(config map alias Map-Lev1-GTP-Filter) # rule add pass ipfrag all-frag-no-first** **(config map alias Map-Lev1-GTP-Filter) # to VP-Filter** **(config map alias Map-Lev1-GTP-Filter) # from 1/1/x6,1/1/x8,1/1/x10,1/1/x12** **(config map alias Map-Lev1-GTP-Filter) # exit** **(config) #**
9.	Create a first level map that directs GTP traffic from a network port to the virtual port for flow sampling. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias Map-Lev1-GTP-Sample** **(config map alias Map-Lev1-GTP-Sample) # type firstLevel byRule** **(config map alias Map-Lev1-GTP-Sample) # roles replace admin to owner_roles** **(config map alias Map-Lev1-GTP-Sample) # rule add pass portdst 2123 bidir** **(config map alias Map-Lev1-GTP-Sample) # rule add pass portdst 2152 bidir** **(config map alias Map-Lev1-GTP-Sample) # rule add pass ipfrag all-frag-no-first** **(config map alias Map-Lev1-GTP-Sample) # to VP-Sample** **(config map alias Map-Lev1-GTP-Sample) # from 1/1/q1** **(config map alias Map-Lev1-GTP-Sample) # exit** **(config) #**

Step	Description	Command
10.	Configure a second level map for GTP flow filtering for version 1 traffic coming from the virtual port for flow filtering and going to the port group for version 1.	**(config) # map alias Map-Lev2-GTP-Filter-Version1** **(config map alias Map-Lev2-GTP-Filter-Version1) # type secondLevel flowFilter** **(config map alias Map-Lev2-GTP-Filter-Version1) # roles replace admin to owner_roles** **(config map alias Map-Lev2-GTP-Filter-Version1) # use gsop GSOP-Filter** **(config map alias Map-Lev2-GTP-Filter-Version1) # flowrule add pass gtp imsi * version 1** **(config map alias Map-Lev2-GTP-Filter-Version1) # to PG-Filter-Version1** **(config map alias Map-Lev2-GTP-Filter-Version1) # from VP-Filter** **(config map alias Map-Lev2-GTP-Filter-Version1) # exit** **(config) #**
11.	Configure another second level map for GTP flow filtering for version 2 traffic coming from the virtual port for flow filtering and going to the port group for version 2.	**(config) # map alias Map-Lev2-GTP-Filter-Version2** **(config map alias Map-Lev2-GTP-Filter-Version2) # type secondLevel flowFilter** **(config map alias Map-Lev2-GTP-Filter-Version2) # roles replace admin to owner_roles** **(config map alias Map-Lev2-GTP-Filter-Version2) # use gsop GSOP-Filter** **(config map alias Map-Lev2-GTP-Filter-Version2) # flowrule add pass gtp imsi * version 2** **(config map alias Map-Lev2-GTP-Filter-Version2) # to PG-Filter-Version2** **(config map alias Map-Lev2-GTP-Filter-Version2) # from VP-Filter** **(config map alias Map-Lev2-GTP-Filter-Version2) # exit** **(config) #**
12.	Configure a second level map for GTP flow sampling traffic coming from the virtual port for flow sampling and going to the port group for flow sampling.	**(config) # map alias Lev2-GTP-Sample** **(config map alias Lev2-GTP-Sample) # type secondLevel flowSample** **(config map alias Lev2-GTP-Sample) # roles replace admin to owner_roles** **(config map alias Lev2-GTP-Sample) # use gsop GSOP-Sample** **(config map alias Lev2-GTP-Sample) #**

Step	Description	Command
		flowsample add gtp percentage 80 **(config map alias Lev2-GTP-Sample) # to PG-Sample** **(config map alias Lev2-GTP-Sample) # from VP-Sample** **(config map alias Lev2-GTP-Sample) # exit** **(config) #**
13.	Add a shared collector for any unmatched traffic from the virtual port for flow filtering.	**(config) # map-scollector alias Collector-Filter** **(config map-scollector alias Collector-Filter) # roles replace admin to owner_roles** **(config map-scollector alias Collector-Filter) # from VP-Filter** **(config map-scollector alias Collector-Filter) # collector GTP-Collector-Filter** **(config map-scollector alias Collector-Filter) # exit** **(config) #**
14.	Add a shared collector for any unmatched traffic from the virtual port for flow sampling.	**(config) # map-scollector alias Collector-Sample** **(config map-scollector alias Collector-Sample) # roles replace admin to owner_roles** **(config map-scollector alias Collector-Sample) # from VP-Sample** **(config map-scollector alias Collector-Sample) # collector GTP-Collector-Sample** **(config map-scollector alias Collector-Sample) # exit** **(config) #**
15.	Display the configuration for this example.	**(config) # show map brief** **(config) # show gigastream** **(config) # show port-group** **(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show map** **(config) # show map brief** **(config) # show load-balance port-group stats all**

Example 7: APN for GigaSMART GTP Whitelisting, APN and QCI for GigaSMART GTP Flow Sampling

Example 7 specifies APN patterns for GTP whitelisting and GTP flow sampling. It also specifies QCI for GTP flow sampling.

In Example 7, traffic from network ports go to the two first level maps (gtp_to_v1_c and gtp_to_v1_u) and then to the virtual port (v1).

In the whitelist map, if there is a match to the APN pattern and if the IMSI is present in the whitelist (IMSI), packets are forwarded to a tool port.

If there is not a match to an IMSI in the whitelist, the traffic is flow sampled based on the APN pattern and QCI value in the flow sampling map. Accepted packets are forwarded to the same tool port as specified in the whitelist map. Only 50% of traffic with QCI 5 is sent to the tool port.

Any unmatched traffic goes to a shared collector that sends it to a different tool port.

The following table provides step-by-step configuration for AFM-based GTP whitelisting and flow sampling using APN and QCI.

Step	Description	Command
1.	Configure a network port and two tool ports and enable them.	**(config) # port 22/3/x3 type network** **(config) # port 22/4/x18 type tool** **(config) # port 22/4/x19 type tool** **(config) # port 22/3/x3 params admin enable** **(config) # port 22/4/x18 params admin enable** **(config) # port 22/4/x19 params admin enable**
2.	Configure a GigaSMART group and associate it with two GigaSMART engine ports.	**(config) # gsgroup alias gsg2 port-list 22/2/e1,22/2/e2**
3.	Create a virtual port.	**(config) # vport alias v1 gsgroup gsg2**
4.	Configure two first level maps, one for control traffic and one for user traffic.	**(config) # map alias gtp_to_v1_c** **(config map alias gtp_to_v1_c) # type firstLevel byRule** **(config map alias gtp_to_v1_c) # roles replace admin to owner_roles** **(config map alias gtp_to_v1_c) # param traffic control** **(config map alias gtp_to_v1_c) # rule add pass portdst 2123 bidir**

Step	Description	Command
		(config map alias gtp_to_v1_c) # rule add pass portdst 2122 bidir **(config map alias gtp_to_v1_c) # to v1** **(config map alias gtp_to_v1_c) # from 22/3/x3** **(config map alias gtp_to_v1_c) # exit** **(config) #** **(config) # map alias gtp_to_v1_u** **(config map alias gtp_to_v1_u) # type firstLevel byRule** **(config map alias gtp_to_v1_u) # roles replace admin to owner_roles** **(config map alias gtp_to_v1_u) # rule add pass portdst 2152 bidir** **(config map alias gtp_to_v1_u) # rule add pass ipfrag all-frag-no-first** **(config map alias gtp_to_v1_u) # to v1** **(config map alias gtp_to_v1_u) # from 22/3/x3** **(config map alias gtp_to_v1_u) # exit** **(config) #**
5.	Associate the GigaSMART group to the active GTP whitelist.	**(config) # gsparams gsgroup gsg2 gtp-whitelist add IMSI**
6.	Configure the GigaSMART operation for GTP whitelisting.	**(config) #** gsop alias gtp-corelate_gsg_wl flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsg2
7.	Configure a second level map for GTP whitelisting, the whitelist map. If there is a match to the APN pattern and if the IMSI is present in the whitelist (IMSI), packets are forwarded to a tool port.	**(config) # map alias GTP-whitelist** **(config map alias GTP-whitelist) # type secondLevel flowWhitelist** **(config map alias GTP-whitelist) # roles replace admin to owner_roles** **(config map alias GTP-whitelist) # use gsop gtp-corelate_gsg_wl** **(config map alias GTP-whitelist) # whitelist add gtp apn *mobile.com*** **(config map alias GTP-whitelist) # to 22/4/x18** **(config map alias GTP-whitelist) # from v1** **(config map alias GTP-whitelist) # exit** **(config) #**
8.	Configure the GigaSMART operation for GTP flow sampling.	**(config) #** gsop alias gtp-corelate_gsg_fs flow-ops gtp-flowsample lb app gtp metric hashing key imei port-list gsg2

Step	Description	Command
9.	Configure a second level map for GTP flow sampling, the flow sampling map. If there is not a match to an IMSI in the whitelist, the traffic flow is sampled based on the APN pattern in this map. Accepted packets are forwarded to the same tool port as specified in the whitelist map.	**(config) # map alias from_vp_fs1** **(config map alias from_vp_fs1) # type secondLevel flowSample** **(config map alias from_vp_fs1) # roles replace admin to owner_roles** **(config map alias from_vp_fs1) # use gsop gtp-corelate_gsg_fs** **(config map alias from_vp_fs1) # flowsample add gtp apn *ims* qci 5 percentage 50** **(config map alias from_vp_fs1) # flowsample add gtp ims* percentage 100** **(config map alias from_vp_fs1) # to 22/4/x18** **(config map alias from_vp_fs1) # from v1** **(config map alias from_vp_fs1) # exit** **(config) #**
10.	Add a shared collector for any unmatched traffic from the virtual port and send it to a different tool port.	**(config) # map-scollector alias from_vp_scoll** **(config map-scollector alias from_vp_scoll) # roles replace admin to owner_roles** **(config map-scollector alias from_vp_scoll) # from v1** **(config map-scollector alias from_vp_scoll) # collector 22/4/x19** **(config map-scollector alias from_vp_scoll) # exit** **(config) #**
11.	Display the session table.	**(config) # show gsgroup flow-ops-report alias gsg2 type flow-filtering any**

Only 20 characters of the APN pattern are displayed in the session table. A plus sign (+) indicates that there are more characters. In the first USER row, the ims* suffix in the flow sampling map matched. In the second USER row, the *mobile.com* prefix and suffix in the whitelisting map matched.

Example 8: Interfaces for GigaSMART GTP Whitelisting and GTP Flow Sampling

You can configure the interfaces **S1U, S5S8U, N3, and N9** for interface based filtering. For interface based filtering, you must configure the node role as **Standalone UPN** only.

whitelist add gtp interface <4g_interface/5g_interface>

where the values of <4g_interface> are S1U for interface S1U and S5S8U for interface S5/S8-U. The values of <5g_interface> are N9 for N9-Access interface and N3 for N3-Access interface.

The following table provides step-by-step configuration for interface-based GTP whitelisting and flow sampling using GigaSMART.

Step	Description	Command
1	Configure interface based filtering Send traffic from the interface N9 only for all subscribers to the tool (Example: port group "pgLB"), discard for all the others. **Note**: In the Standalone UPN mode, the RAN whitelisting is not supported. Hence the rule “whitelist add gtp type all” is not supported. Instead, you can configure the rule “whitelist add gtp type imsi”.	**(config) # map alias SA-UPN-WL-ITFC** **(config map alias SA-UPN-WL-ITFC) # use gsop WL-Lb** **(config map alias SA-UPN-WL-ITFC) # whitelist add gtp interface N9** **(config map alias SA-UPN-WL-ITFC) # to pgLB** **(config map alias SA-UPN-WL-ITFC) # from vp 1** **(config map alias SA-UPN-WL-ITFC) # exit**

flowsample add gtp <imsi|imei|msisdn <number[*]>> | interface <4g_interface/5g_ interface> percentage <>

where the values of <4g_interface> are S1U for interface S1U and S5S8U for interface S5/S8-U. The values of <5g_interface> are N9 for N9 interface and N3 for N3 interface.

Step	Description	Command
1	Configure interface based filtering Send traffic from the interface S1U only for all subscribers to the tool (Example: port group "pgLB"), discard for all the others.	**(config) # map alias SA-UPN-WL-ITFC** **(config map alias SA-UPN-WL-ITFC) # use gsop FS_Lb** **(config map alias SA-UPN-WL-ITFC) # flowsample add gtp imsi * interface S1U percentage 100** **(config map alias SA-UPN-WL-ITFC) # to pgLB** **(config map alias SA-UPN-WL-ITFC) # from vp 1** **(config map alias SA-UPN-WL-ITFC) # exit**

Flow-Ops for Unsupported Interfaces

The Flow-ops table displays statistics of the interfaces not supported by the GTP correlation engine.

```
Interface            Pkts
===============      ===========
 S2A                    0
 S3                     0
 S4                     0
 S11U                   0
 Unknown                0
```

Flow-Ops for PFCP Node Related Messages

```
PFCP messages stats
===================
Msg Type           Pkt Count
====================================
Heart Beat Req          24
Heart Beat Rsp               24
PFD Mgmt Req                 0
PFD Mgmt Rsp                 0
Asso Setup Req               3
Asso Setup Rsp               3
Asso Upd Req                 0
Asso Upd Rsp                 0
Asso Release Req             1
Asso Release Rsp             1
Ver Not Support Rsp          0
Node Report Req              0
Node Report Rsp              0
Sess Set Del Req             0
Sess Set Del Rsp             0
Reserve Msg Type 16_to_49    0
Reserve Msg Type 58_to_255   0
PFCP bundle messages         0
```

GigaSMART GTP Overlap Flow Sampling Maps

Starting in software version 4.8, GTP overlap flow sampling maps combine GTP whitelisting and GTP flow sampling maps into a new GTP overlap flow sampling map group, which allows for selected traffic to be sent to multiple destinations simultaneously.

In this scenario, once traffic matches a map, it will be sent to the destination for that map. However, the matched traffic will also be evaluated by subsequent maps and, if a match occurs, it will be sent to each of the destinations pointed to by the subsequent maps.

Example 1: GTP Overlap Mode

Example 1 is a GTP overlap flow sampling map example.

In Example 1, traffic from a single network port goes to a single first level map (mapLevel1-GTP) which directs GTP-Control and GTP-User traffic to virtual port (VP31). Traffic from VP31 is replicated to two GTP whitelisting maps (WLMAP1 and WLMAP2) and two GTP flow sampling maps (FSMAP1 and FSMAP2), which then forward accepted traffic to the final port-group destinations, pg1 and pg2, for load balancing (refer to Figure 23GTP Overlap Mode Example 1).

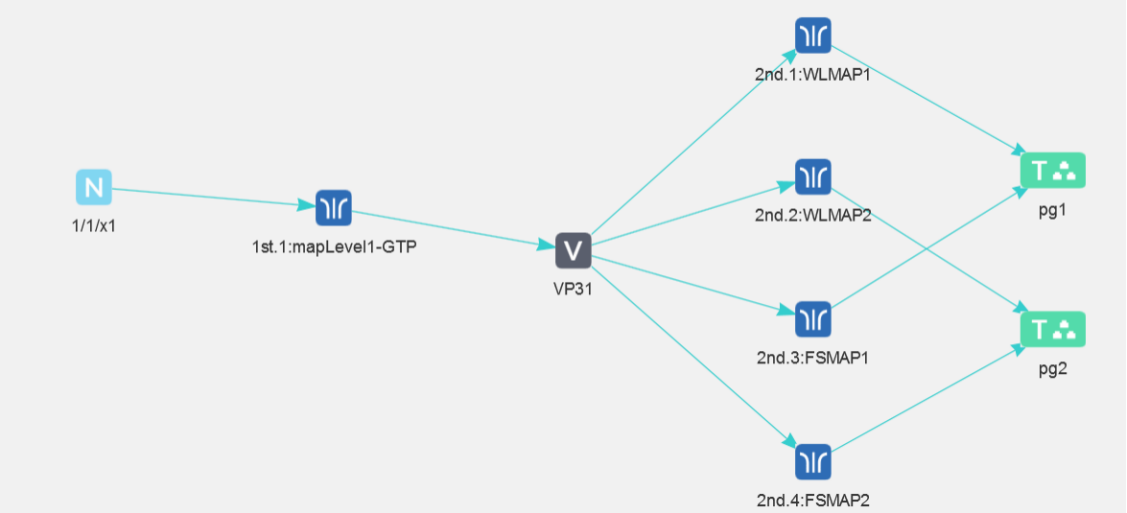


Figure 23 *GTP Overlap Mode Example 1*

NOTE: In Example 1, the tool ports and GigaStream in the port group are on the same node as the GigaSMART group and GigaSMART operation.

Within each GTP whitelisting and flow sampling pair, if there is not a match to an IMSI in the whitelist map, the traffic flow is sampled based on the rules in the flow sampling map. The flow sampling rules specify IMSI, IMEI, and MSISDN numbers, as well as the percentage to sample.

Within each map pair, packets are then accepted or rejected. Accepted packets are forwarded to the port groups for load balancing. Rejected packets are dropped.

Use the steps in the following CLI table to configure Example 1.

Step	Description	Command
1.	Create GigaStream that will be part of the port groups.	**(config) # gigastream alias gs1 port-list 1/1/x16..x17** **(config) # gigastream alias gs2 port-list 1/1/x1..x2**
2.	Create port groups and specify the tool ports.	**(config) # port-group alias pg1 port-list 1/1/x6..x7** **(config) # port-group alias pg2 port-list 1/1/x18..x19**
3.	Assign GigaStream to the port groups.	**(config) # port-group alias pg1 gigastream-list gs1** **(config) # port-group alias pg2 gigastream-list gs2**
4.	Enable load balancing on the port groups.	**(config) # port-group alias pg1 smart-lb enable** **(config) # port-group alias pg2 smart-lb enable**
5.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias GS31 port-list 1/3/e1**
6.	Create a virtual port. Notes: • You must specify **gtp-overlap** mode when configuring a virtual port for GTP overlap flow sampling. • When the vport mode is changed from overlap to non-overlap mode, indeterministic behavior will be in the traffic. Hence, it is recommended to do a cluster or chassis reload for the configuration to take effect.	**(config) # vport alias VP31 gsgroup GS31 mode gtp-overlap**
7.	Configure the first level map. **NOTE:** In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic. **NOTE:** Configure your First-level map before configuring the second-level maps and collector and then the map	**(config) # map alias mapLevel1-GTP** **(config map alias mapLevel1-GTP) # type firstLevel byRule** **(config map alias mapLevel1-GTP) # rule add pass portdst 2123 bidir** **(config map alias mapLevel1-GTP) # rule add pass portdst 2152 bidir** **(config map alias mapLevel1-GTP) # to VP31** **(config map alias mapLevel1-GTP) # from 1/1/x1**

Step	Description	Command
	group.	**(config map alias mapLevel1-GTP) # exit** **(config) #**
8.	Create the GTP whitelist.	**(config) # apps gtp-whitelist alias Whitelist create**
9.	Fetch whitelist files from a specified location to populate the GTP whitelist.	**(config) # apps gtp-whitelist alias Whitelist fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file1.txt** **(config) # apps gtp-whitelist alias Whitelist fetch add http://10.1.1.100/tftpboot/myfiles/MyIMSIs_file2.txt**
10.	Associate the GigaSMART group to the GTP whitelist.	**(config) # gsparams gsgroup GS31 gtp-whitelist add Whitelist**
11.	Configure the GigaSMART operation for GTP whitelisting. **NOTE:** GigaSMART operations used in GTP overlap flow sampling map mode must be enabled for load balancing.	**(config) #** gsop alias gtp-overlapwhitelist1 flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list GS31
12.	Configure the GigaSMART operation for GTP flow sampling.	**(config) #** gsop alias gtp-overlapsample1 flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list GS31
13.	Configure the first second level GTP overlap map for GTP whitelisting. If there is a match to an IMSI in the whitelist for GTP version 1 traffic, it is then forwarded to load balancing port group **pg1**.	**(config) # map alias WLMAP1** **(config map alias WLMAP1) # type secondLevel flowWhitelist-ol** **(config map alias WLMAP1) # use gsop gtp-whitelist** **(config map alias WLMAP1) # whitelist add gtp version 1** **(config map alias WLMAP1) # to pg1** **(config map alias WLMAP1) # from VP31** **(config map alias WLMAP1) # exit** **(config) #**
14.	Configure a second level map for GTP flow sampling, the flow sampling map. If there is not a match to an IMSI in the whitelist, the traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to load balancing port group **pg1**.	**(config) # map alias FSMAP1** **(config map alias FSMAP1) # type secondLevel flowSample-ol** **(config map alias FSMAP1) # use gsop gtp-overlapsample1** **(config map alias FSMAP1) # flowsample add gtp imsi 3102609834* imei 35609506* percentage 20** **(config map alias FSMAP1) # to pg1**

Step	Description	Command
		(config map alias FSMAP1) # from VP31 **(config map alias FSMAP1) # exit** **(config) #**
15.	Configure the next second level GTP overlap map for GTP whitelisting. If there is a match to an IMSI in the whitelist for GTP version 2 traffic, it is then forwarded to load balancing port group **pg2**.	**(config) # map alias WLMAP2** **(config map alias WLMAP2) # type secondLevel flowWhitelist-ol** **(config map alias WLMAP2) # use gsop gtp-whitelist** **(config map alias WLMAP2) # whitelist add gtp version 2** **(config map alias WLMAP2) # to pg2** **(config map alias WLMAP2) # from VP31** **(config map alias WLMAP2) # exit** **(config) #**
16.	Configure the next second level map for GTP flow sampling. If there is not a match to an IMSI in the whitelist as evaluated by the second level GTP whitelisting map **WLMAP2**, the traffic flow is sampled based on the rules in this map. Accepted packets are forwarded to load balancing port group **pg2**.	**(config) # map alias FSMAP2** **(config map alias FSMAP2) # type secondLevel flowSample-ol** **(config map alias FSMAP2) # use gsop gtp-overlapsample1** **(config map alias FSMAP2) # flowsample add gtp imsi 3102609835* imei 35609507* percentage 20** **(config map alias FSMAP2) # to pg2** **(config map alias FSMAP2) # from VP31** **(config map alias FSMAP2) # exit** **(config) #**
17.	Configure a map group. Add the GTP whitelisting and the two GTP flow sampling maps configured in previous steps. **NOTE:** For overlap mode, you must configure the maps in a sequential order in the map-group. **NOTE:** You must configure the map group for GTP overlap flow sampling. Make sure to include all the whitelisting and flow sampling maps in your configuration.	**(config) # map-group alias OverlapMap map-list WLMAP1,WLMAP2,FSMAP1,FSMAP2**

Step	Description	Command
18.	Display the configuration for this example.	**(config) # show port-group** **(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show gsparams** **(config) # show map** **(config) # show map-group**

GigaSMART GTP Scaling

GTP can be scaled as follows:

- GigaSMART Cards in GigaVUE-HC3
- GTP Engine Grouping

GigaSMART Cards in GigaVUE-HC3

Required License: GTP Filtering & Correlation

A total of four GigaSMART SMT-HC3-C05 line cards are supported on a single GigaVUE-HC3 node. This provides a total of eight GigaSMART engine ports, which increases the amount of GigaSMART processing available on the GigaVUE-HC3.

The increased number of GigaSMART line cards in the GigaVUE-HC3 can be used by the following GTP applications: GTP flow filtering, GTP flow sampling, and GTP whitelisting.

GTP Engine Grouping

Required License: GTP Filtering & Correlation

A GigaSMART group (gsgroup) associated with GTP applications can have multiple GigaSMART engine port members. Up to four engine ports can be combined to form a GTP engine group. The engine group provides higher capacity to GTP applications by load balancing GTP user-data plane (GTP-u) traffic among the members of the group. Grouping multiple GigaSMART engine ports increases the effective throughput for GTP applications.

GTP Engine Grouping Configuration Example

This is an example of a GTP engine group consisting of two engine ports on a GigaVUE-HC3 node. This example includes a GigaSMART operation for GTP flow filtering.

The following table provides step-by-step commands for GTP engine configuration for flow filtering and traffic control.

Step	Description	Command
1.	Configure ports as follows: • one network type of port. This will be used as the **from** attribute in two first level maps in Step 5 and Step 6. • one tool type of port for the **to** attribute in a second level flow filtering map in Step 7. • one tool type of port for the **to** attribute in a shared collector map in Step 8. Then administratively enable the ports.	**(config) # port 22/3/x3 type network** **(config) # port 22/3/x1 type tool** **(config) # port 22/1/x11 type tool** **(config) # port 22/3/x3 params admin enable** **(config) # port 22/3/x1 params admin enable** **(config) # port 22/1/x11 params admin enable**
2.	Configure a GigaSMART group and associate it with two GigaSMART engine ports, to form the GTP engine group. The GigaSMART group will be used in Step 3 and Step 4.	**(config) # gsgroup alias gsg2 port-list 22/2/e1,23/1/e2**
3.	For GTP flow filtering, configure a flow filtering GigaSMART operation and assign it to the GigaSMART group. The gsop will be used in the second level flow filtering map in Step 7.	**(config) # gsop alias gtp_gsg2 flow-ops flow-filtering gtp port-list gsg2**
4.	Configure a virtual port and assign it to the same GigaSMART group. This virtual port will be used as the **to** attribute in the first level maps in Step 5 and Step 6, as the **from** attribute in the second level map in Step 7, and as the **from** attribute in the shared collector map in Step 8.	**(config) # vport alias vp1 gsgroup gsg2**
5.	Create a first level map that directs GTP control traffic from the physical network port to the virtual port created in Step 4. **Note**: In the rule, 2123 is GTP-c traffic. This map, with the **param traffic control** attribute, identifies the GTP-c control traffic needed for GTP engine grouping. **Note**: The order of configuration is	**(config) # map alias gtp_to_vp1-c** **(config map alias gtp_to_vp1-c) # type firstLevel byRule** **(config map alias gtp_to_vp1-c) # roles replace admin to owner_roles** **(config map alias gtp_to_vp1-c) # param traffic control** **(config map alias gtp_to_vp1-c) # rule add pass portdst 2123 bidir** **(config map alias gtp_to_vp1-c) # to vp1**

Step	Description	Command
	important. Configure **param traffic control** before any map rules.	**(config map alias gtp_to_vp1-c) # from 22/3/x3** **(config map alias gtp_to_vp1-c) # exit** **(config) #**
6.	Create another first level map that directs GTP user traffic from the physical network port to the virtual port created in Step 4. **Note**: In the rule, 2152 is GTP-u traffic. GTP-u traffic corresponding to the same GTP-c traffic will be sent to the same virtual port.	**(config) # map alias gtp_to_vp1** **(config map alias gtp_to_vp1) # type firstLevel byRule** **(config map alias gtp_to_vp1) # roles replace admin to owner_roles** **(config map alias gtp_to_vp1) # rule add pass portdst 2152 bidir** **(config map alias gtp_to_vp1) # rule add pass ipfrag all-frag-no-first** **(config map alias gtp_to_vp1) # to vp1** **(config map alias gtp_to_vp1) # from 22/3/x3** **(config map alias gtp_to_vp1) # exit** **(config) #**
7.	Create a second level map for GTP flow filtering that takes traffic from the virtual port, applies the flow filtering GigaSMART operation, matches IMEIs and version specified by the flow rule, and sends matching traffic to a tool port.	**(config) # map alias from_vp1** **(config map alias from_vp1) # type secondLevel flowFilter** **(config map alias from_vp1) # roles replace admin to owner_roles** **(config map alias from_vp1) # use gsop gtp_gsg2** **(config map alias from_vp1) # flowrule add pass gtp imei * version 2** **(config map alias from_vp1) # to 22/3/x1** **(config map alias from_vp1) # from vp1** **(config map alias from_vp1) # exit** **(config) #**
8.	Add a shared collector for any unmatched traffic from the virtual port and send it to a different tool port than in Step 7.	**(config) # map-scollector alias from_vp1_scoll** **(config map-scollector alias from_vp1_scoll) # roles replace admin to owner_roles** **(config map-scollector alias from_vp1_scoll) # from vp1** **(config map-scollector alias from_vp1_scoll) # collector 22/1/x11** **(config map-scollector alias from_vp1_scoll) # exit** **(config) #**

GTP Engine Grouping Configuration Complex Example

This is a more complex example of GTP engine grouping than the previous example. This example has four engine ports on two GigaSMART line cards on the same GigaVUE-HC3 node. The GigaSMART line cards are in slots 1 and 3.

The GigaVUE-HC3 node is the cluster leader of a two-node out-of-band cluster.

This example includes GigaSMART operations for GTP flow filtering with load balancing, GTP flow sampling with load balancing, and GTP whitelisting. The whitelist must be associated with the GigaSMART group on the leader, the GigaVUE-HC3.

The following table illustrates step-by-step configuration for clustered GTP engine grouping and advanced traffic handling.

Step	Description	Command
1.	Configure ports on the GigaVUE-HC3 as follows: • one network type of port. This will be used as the **from** attribute in two first level maps in Step 11 and Step 12. • twelve tool type of ports. There are four tool ports in each of three port groups used for load balancing. The port groups will be created in Step 6. • five tool type of ports for a GigaStream that will be created in Step 2. • two tool type of ports for another GigaStream that will be created in Step 2. Then administratively enable the ports.	**(config) # port 23/2/c3 type network** **(config) # port 23/3/x1..x4 type tool** **(config) # port 23/3/x9..x12 type tool** **(config) # port 23/3/x13..x16 type tool** **(config) # port 23/3/x20..x24 type tool** **(config) # port 23/2/c1..c2 type tool** **(config) # port 23/2/c3 params admin enable** **(config) # port 23/3/x1..x4 params admin enable** **(config) # port 23/3/x9..x12 params admin enable** **(config) # port 23/3/x13..x16 params admin enable** **(config) # port 23/3/x20..x24 params admin enable** **(config) # port 23/2/c1..c2 params admin enable**
2.	On the GigaVUE-HC3, configure one GigaStream using five tool ports. This will be used as the to attribute in the map in Step 11. Configure another GigaStream to be used in the stack link between the GigaVUE-HC3 that will be created in Step 5.	**(config) # gigastream alias hc3-gs-1 port-list 23/3/x20..x24 params hash advanced** **(config) # gigastream alias hc3-80g port-list 23/2/c1..c2 params hash advanced**
3.	Configure ports on the GigaVUE-HC1-Plus as follows:	**(config) # port 33/2/q1..q2 type tool** **(config) # port 33/3/x11 type tool**

Step	Description	Command
	• two tool type of ports for a GigaStream that will be created in Step 4. • one tool type of port that will be used as the **to** attribute in a map in Step 11. • four tool type of ports for a GigaStream that will be created in Step 4. Then administratively enable the ports.	**(config) # port 33/2/x20..x24 type tool** **(config) # port 33/2/q1..q2 params admin enable** **(config) # port 33/3/x11 params admin enable** **(config) # port 33/2/x20..x24 params admin enable**
4.	On the GigaVUE-HC1-Plus, configure a GigaStream using two tool ports. This will be used in the stack link created in Step 5. Configure another GigaStream using four tool ports. This will be used in the shared collector in Step 17.	**(config) # gigastream alias hc3-80g port-list 33/2/q1..q2 params hash advanced** **(config) #s gigastream alias hc3-gs-4 port-list 33/2/x20..x24 params hash advanced**
5.	Configure the stack link between the GigaVUE-HC1-Plus and GigaVUE-HC3.	**(config) # stack-link alias hc1p-hc3 between gigastreams hc3-80g and hc1p-80g**
6.	Create three port groups and specify four tool ports each, for load balancing. Also, enable load balancing on each port group. The port groups, hc3-pg-1 and hc3-pg-2, will be used as the **to** attribute in two second level flow sampling maps in Step 14 and Step 15. The port group, hc3-q2x24-1-4, will be used as the **to** attribute in a second level flow filtering map in Step 16.	**(config) # port-group alias hc3-q2x24-1-4** **(config port-group alias hc3-q2x24-1-4) # port-list 23/4/x1..x4** **(config port-group alias hc3-q2x24-1-4) # smart-lb enable** **(config port-group alias hc3-q2x24-1-4) # exit** **(config) # port-group alias hc3-pg-1** **(config port-group alias hc3-pg-1) # port-list 23/4/x9..x12** **(config port-group alias hc3-pg-1) # smart-lb enable** **(config port-group alias hc3-pg-1) # exit** **(config) # port-group alias hc3-pg-2** **(config port-group alias hc3-pg-2) # port-list 23/4/x13..x16** **(config port-group alias hc3-pg-2) # smart-lb enable** **(config port-group alias hc3-pg-2) # exit**
7.	Configure a GigaSMART group and associate it with four GigaSMART engine ports, two in slot 1 and two in	**(config) # gsgroup alias hc3scale-3engines-slots1and3 port-list 23/1/e1,23/1/e2,23/3/e1,23/3/e2**

Step	Description	Command
	slot 3, to form the GTP engine group. The GigaSMART group will be used in Step 8, Step 9, and Step 10.	
8.	Associate the GigaSMART group to an existing GTP whitelist. **Note**: The whitelist must be associated with the GigaSMART group on the leader, which is the GigaVUE-HC3 in this example.	**(config) # gsparams gsgroup hc3scale-4engines-slots1and3 gtp-whitelist add 500-1**
9.	For GTP flow filtering, configure a flow filtering GigaSMART operation, specify load balancing, and assign the GigaSMART operation to the GigaSMART group. The hc3-scale-ff-lb gsop will be used in the second level flow filtering map in Step 16. For GTP flow sampling, configure a flow sampling GigaSMART operation, specify load balancing, and assign the GigaSMART operation to the GigaSMART group. The hc3-scale-fs-lb gsop will be used in the two second level flow sampling maps in Step 14 and Step 15. For GTP whitelisting, configure a whitelisting GigaSMART operation, and assign the GigaSMART operation to the GigaSMART group. The hc3-scale-wl gsop will be used in the second level whitelisting map in Step 13. (This GigaSMART operation is not load balanced.)	**(config) # gsop alias hc3-scale-ff-lb flow-ops flow-filtering gtp lb app gtp metric hashing key imsi port-list hc3scale-4engines-slots1and3** **(config) # gsop alias hc3-scale-fs-lb flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list hc3scale-4engines-slots1and3** **(config) # gsop alias hc3-scale-wl flow-ops gtp-whitelist port-list hc3scale-4engines-slots1and3**
10.	Configure a virtual port and assign it to the same GigaSMART group. This virtual port will be used as the **to** attribute in the first level maps in Step 11 and Step 12, as the **from** attribute in the second level maps in Step 13, Step 14, Step 15, Step 16, and as the **from** attribute in the shared collector in Step 17.	**(config) # vport alias vp-hc3scale-4engines-slots1and3 gsgroup hc3scale-4engines-slots1and3**
11.	Create a first level map that directs GTP control traffic from the physical network port to the virtual port	**(config) # map alias to_hc3_gtpc** **(config map alias to_hc3_gtpc) # type firstLevel byRule**

Step	Description	Command
	created in Step 10. **Note**: In the rule, 2123 is GTP-c traffic. This map, with the **param traffic control** attribute, identifies the GTP-c control traffic needed for GTP engine grouping. **Note**: The order of configuration is important. Configure **param traffic control** before any map rules. In addition to the virtual port, traffic is also sent to a GigaStream and a tool port.	**(config map alias to_hc3_gtpc) # roles replace admin to owner_roles** **(config map alias to_hc3_gtpc) # param traffic control** **(config map alias to_hc3_gtpc) # rule add pass portdst 2123 bidir** **(config map alias to_hc3_gtpc) # to hc3-gs-1,vp-hc3scale-4engines-slots1and3,33/3/x11** **(config map alias to_hc3_gtpc) # from 23/3/q6** **(config map alias to_hc3_gtpc) # exit** **(config) #**
12.	Create another first level map that directs GTP user traffic from the physical network port to the virtual port created in Step 10. **Note**: In the rule, 2152 is GTP-u traffic. GTP-u traffic corresponding to the same GTP-c traffic will be sent to the same virtual port.	**(config) # map alias to_hc3_gtpu_1** **(config map alias to_hc3_gtpu_1) # type firstLevel byRule** **(config map alias to_hc3_gtpu_1) # roles replace admin to owner_roles** **(config map alias to_hc3_gtpu_1) # rule add pass portdst 2152 bidir** **(config map alias to_hc3_gtpu_1) # rule add pass ipfrag all-frag-no-first** **(config map alias to_hc3_gtpu_1) # to vp-hc3scale-4engines-slots1and3** **(config map alias to_hc3_gtpu_1) # from 23/7/q6** **(config map alias to_hc3_gtpu_1) # exit** **(config) #**
13.	Configure a second level map for GTP whitelisting, the whitelist map, that takes traffic from the virtual port, applies the whitelisting GigaSMART operation, and sends traffic to the remoteGigaVUE-HC1-Plus node through a GigaStream.	**(config) # map alias from_hc3_wl** **(config map alias from_hc3_wl) # type secondLevel flowWhitelist** **(config map alias from_hc3_wl) # roles replace admin to owner_roles** **(config map alias from_hc3_wl) # use gsop hc3-scale-w** **(config map alias from_hc3_wl) # to hc3-gs-1** **(config map alias from_hc3_wl) # from vp-hc3scale-4engines-slots1and3** **(config map alias from_hc3_wl) # exit** **(config) #**
14.	Configure a second level map for GTP flow sampling. This is the first of two	**(config) # map alias from_hc3_fs_v2**

Step	Description	Command
	flow sampling maps. This map filters for version 2. It takes traffic from the virtual port and applies the flow sampling GigaSMART operation. Traffic flow is sampled based on the flow sampling rule in this map. Accepted packets are forwarded to load balancing port group hc3-pg-2.	**(config map alias from_hc3_fs_v2) # type secondLevel flowSample** **(config map alias from_hc3_fs_v2) # roles replace admin to owner_roles** **(config map alias from_hc3_fs_v2) # use gsop hc3-scale-fs-lb** **(config map alias from_hc3_fs_v2) # flowsample add gtp imsi 5* version 2 percentage 60** **(config map alias from_hc3_fs_v2) # to hc3-pg-2** **(config map alias from_hc3_fs_v2) # from vp-hc3scale-4engines-slots1and3** **(config map alias from_hd3_fs_v2) # exit** **(config) #**
15.	Configure a second level map for GTP flow sampling. This is the second of two flow sampling maps. This map filters for version 1. It takes traffic from the virtual port and applies the flow sampling GigaSMART operation. Traffic flow is sampled based on the flow sampling rule in this map. Accepted packets are forwarded to load balancing port group hc3-pg-1.	**(config) # map alias from_hc3_fs_v1** **(config map alias from_hc3_fs_v1) # type secondLevel flowSample** **(config map alias from_hc3_fs_v1) # roles replace admin to owner_roles** **(config map alias from_hc3_fs_v1) # use gsop hc3-scale-fs-lb** **(config map alias from_hc3_fs_v1) # flowsample add gtp imsi 5* version 1 percentage 60** **(config map alias from_hc3_fs_v1) # to hc3-pg-1** **(config map alias from_hc3_fs_v1) # from vp-hc3scale-4engines-slots1and3** **(config map alias from_hc3_fs_v1) # exit** **(config) #**
16.	Create a second level map for GTP flow filtering that takes traffic from the virtual port, applies the flow filtering GigaSMART operation, matches IMSIs specified by the flow rule, and sends matching traffic to load balancing port group hc3-q2x24-1-4.	**(config) # map alias from_hc3_ff** **(config map alias from_hc3_ff) # type secondLevel flowFilter** **(config map alias from_hc3_ff) # roles replace admin to owner_roles** **(config map alias from_hc3_ff) # use gsop hc3-scale-ff-** **(config map alias from_hc3_ff) # flowrule add pass gtp imsi *** **(config map alias from_hc3_ff) # to hc3-q2x24-1-4**

Step	Description	Command
		(config map alias from_hc3_ff) # from vp-hc3scale-4engines-slots1and3 **(config map alias from_hc3_ff) # exit** **(config) #**
17.	Add a shared collector for any unmatched traffic from the virtual port and send it to a GigaStream.	**(config) # map-scollector alias s_coll_hc3** **(config map-scollector alias s_coll_hc3) # roles replace admin to owner_roles** **(config map-scollector alias s_coll_hc3) # from vp-hc3scale-4engines-slots1and3** **(config map-scollector alias s_coll_hc3) # collector hc3-gs-4** **(config map-scollector alias s_coll_hc3) # exit** **(config) #**
18.	Display the configuration for this example.	**(config) # show gigastream** **(config) # show stack-link** **(config) # show port-group** **(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show map**

GigaSMART GTP Stateful Session Recovery

GTP sessions can be backed up periodically so they can then be recovered faster after a GigaSMART line card reboot or a node reboot. GTP stateful session recovery provides session persistence for GigaSMART GTP applications, including GTP flow filtering, GTP whitelisting, and GTP flow sampling.

GTP stateful session recovery requires additional memory for storing backups. GigaVUE-HC3 has the required memory. For GigaVUE-OS HD Series, a memory upgrade for control card HCCv2 is available. Contact your Sales representative or authorized partner for the required control cards for GigaVUE-OS HD Series.

Restore and Backup GTP Persistence

To backup and restore the GTP persistence file, let's consider an example of a *gsgroup* with two engines.

Create a Backup of the GTP persistence file

1. Increase the GTP persistence file age and restart-age timeouts:

```
gsparams GSG_GTP gtp-persistence file-age-timeout 1440
gtp-persistence restart-age-time 1440 exit
```

2. Pause the GTP persistence backup. An immediate backup will start.

   ```
   gsgroup alias GSG_GTP gtp-persistence pause 100
   ```

3. Check that the backup has completed for all engines.

   ```
   show gsgroup gtp-persistence all detail
   ```

4. Upload the backup of all engines in the *gsgroup* to a remote server

   ```
   gsgroup alias GSG_GTP gtp-persistence upload
   ```

5. Proceed to make any GTP configurarion changes as required.

Restore the GTP persistence file

1. GTP persistence must be enabled on the gsgroup
2. Increase the GTP persistence file age and restart-age timeouts:

   ```
   gsparams GSG_GTP gtp-persistence file-age-timeout 1440
   gtp-persistence restart-age-time 1440exit
   ```

3. Pause the GTP persistence backup with the skip-backup option.

   ```
   gsgroup alias GSG_GTP gtp-persistence pause 100 skip-backup
   ```

4. Download the GTP persistence file to each engine. Verify if the files match the ones uploaded.

   ```
   gsgroup alias GSG_GTP gtp-persistence download
   ```

5. Reload the GigaSMART cards for the engines to restore the sessions during bootup.

GigaSMART SIP/RTP Correlation

Session Initiation Protocol (SIP) is the dominant method to initiate, maintain, modify, and terminate voice calls in service provider and enterprise networks. Real-time Transport Protocol (RTP) is used to manage the real-time transmission of voice payload across the same networks. Visibility into a subscriber's voice traffic requires the ability to understand the subscriber attributes and stateful information contained within SIP to correlate subscriber-specific RTP traffic so that monitoring tools can achieve an accurate view of the subscriber's traffic on the network.

SIP/RTP Examples

Refer to the following examples:

- SIP/RTP Minimum Configuration Example
- SIP/RTP Load Balancing Example

For details on the CLI commands used in the following examples, refer to the following commands in the reference section:

- apps sip-whitelist
- gsgroup
- gsop
- gsparams
- map
- port-group
- vport

SIP/RTP Minimum Configuration Example

This is a minimum configuration example of SIP/RTP.

Step	Description	Command
1.	Configure ports as follows: • one network type of port. This will be used as the **from** attribute in the first level map. • one tool type of port for the **to** attribute in the second level maps. Then administratively enable the ports.	**(config) # port 1/1/g1 type network** **(config) # port 1/1/g2 type tool** **(config) # port 1/1/g1 params admin enable** **(config) # port 1/1/g2 params admin enable**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine ports.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure GigaSMART parameters for the SIP port list and the RTP port range. **NOTE:** The SIP port list is 5060 by default. The RTP port range must be specified, otherwise all RTP will be dropped.	**(config) # gsparams gsgroup alias gsg1** **(config gsparams gsgroup gsg1) # rtp-port range 2000..3000** **(config gsparams gsgroup gsg1) # sip-portlist 5060** **(config gsparams gsgroup gsg1) # sip-session timeout 30** **(config gsparams gsgroup gsg1) # sip-whitelist add sipwl1** **(config gsparams gsgroup gsg1) # exit** **(config) #**
4.	Configure a GigaSMART operation for either SIP flow sampling or SIP flow whitelisting or both.	**(config) # gsop alias gsop-SIP flow-ops sip-flowsample port-list gsg1** **(config) # gsop alias gsop-sipWL flow-ops sip-whitelist port-list gsg1**
5.	Configure a virtual port and assign it to the same GigaSMART group.	**(config) # vport alias vport1 gsgroup gsg1** **(config) # vport alias vport1 failover-action**

Step	Description	Command
	Then configure a failover action on the virtual port.	**vport-bypass**
6.	Create a first level map.	**(config) # map alias map-Level1** **(config map alias map-Level1) # type firstLevel byRule** **(config map alias map-Level1) # roles replace admin to owner_roles** **(config map alias map-Level1) # rule add pass vlan 2040 bidir** **(config map alias map-Level1) # rule add pass vlan 2030..2035** **(config map alias map-Level1) # to vport1** **(config map alias map-Level1) # from 1/1/g1** **(config map alias map-Level1) # exit** **(config) #**
7.	Create a second level map for SIP whitelisting. Only one SIP whitelist map can be configured.	**(config) # map alias map-sipWL** **(config map alias map-sipWL) # type secondLevel flowWhitelist-sip** **(config map alias map-sipWL) # roles replace admin to owner_roles** **(config map alias map-sipWL) # use gsop gsop-sipWL** **(config map alias map-sipWL) # to 1/1/g2** **(config map alias map-sipWL) # from vport1** **(config map alias map-sipWL) # exit** **(config) #**
8.	Create another second level map for SIP flow sampling. Up to five SIP flow sample maps can be configured.	**(config) # map alias map-sipFS** **(config map alias map-sipFS) # type secondLevel flowSample-sip** **(config map alias map-sipFS) # roles replace admin to owner_roles** **(config map alias map-sipFS) # use gsop gsop-SIP** **(config map alias map-sipFS) # flowsample add sip caller-id 510* percentage 65** **(config map alias map-sipFS) # flowsample add sip caller-id 408* percentage 25** **(config map alias map-sipFS) # to 1/1/g2** **(config map alias map-sipFS) # from vport1** **(config map alias map-sipFS) # exit** **(config) #**

SIP/RTP Load Balancing Example

This is a load balancing configuration example of SIP/RTP. Refer to Figure 24SIP/RTP Load Balancing Topology for the topology.

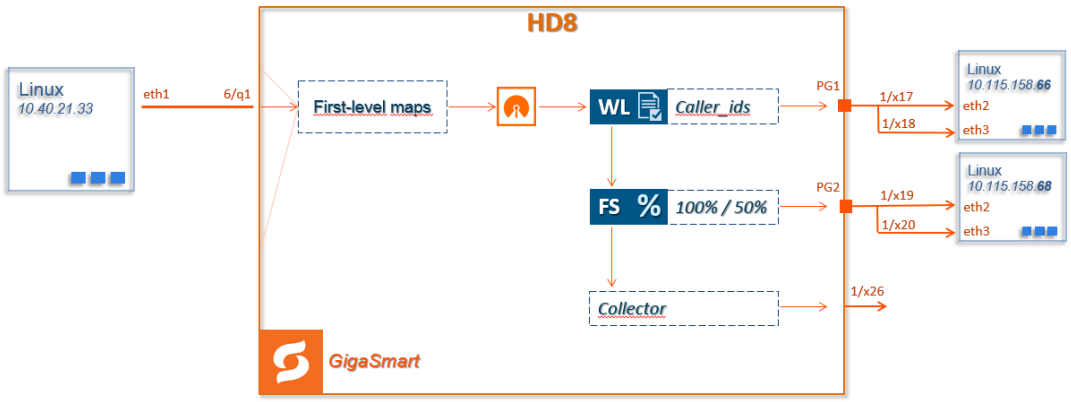


Figure 24 *SIP/RTP Load Balancing Topology*

Step	Description	Command
1.	Configure ports as follows: • one network type of port. This will be used as the **from** attribute in the first level map. • four tool type of ports for the port group port lists • one tool type of port for the **to** attribute in a shared collector map Then administratively enable the ports.	**(config) # port 10/6/q1 type network** **(config) # port 10/1/x17..10/1/x20 type tool** **(config) # port 10/1/x26 type tool** **(config) # port 10/6/q1 params admin enable** **(config) # port 10/1/x17..10/1/x20 params admin enable** **(config) # port 10/1/x26 params admin enable**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine ports.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure GigaSMART parameters for the SIP port list and the RTP port range. **NOTE:** The SIP port list is 5060 by default. The RTP port range must be specified, otherwise all RTP will be dropped.	**(config) # gsparams gsgroup alias gsg1** **(config gsparams gsgroup gsg1) # rtp-port range 2000..3000** **(config gsparams gsgroup gsg1) # sip-portlist 5060** **(config gsparams gsgroup gsg1) # sip-session timeout 30** **(config gsparams gsgroup gsg1) # sip-whitelist add sipwl1** **(config gsparams gsgroup gsg1) # exit** **(config) #**

Step	Description	Command
4.	Configure GigaSMART operations with load balancing for SIP whitelisting and SIP flow sampling.	**(config) # gsop alias sip-wl-lb flow-ops sip-whitelist lb app sip metric hashing key caller-id port-list gsg1** **(config) # gsop alias sip-fs-lb flow-ops sip-flowsample lb app sip metric hashing key caller-id port-list gsg1**
5.	Configure a virtual port and assign it to the same GigaSMART group. Then configure a failover action on the virtual port.	**(config) # vport alias vport1 gsgroup gsg1** **(config) # vport alias vport1 failover-action vport-bypass**
6.	Configure two port groups.	**(config) # port-group alias pg1** **(config port-group alias pg1) # port-list 10/1/x17..x18** **(config port-group alias pg1) # smart-lb enable** **(config port-group alias pg1) # exit** **(config) #** **(config) # port-group alias pg2** **(config port-group alias pg1) # port-list 10/1/x19..x20** **(config port-group alias pg1) # smart-lb enable** **(config port-group alias pg1) # exit** **(config) #**
7.	Create a first level map.	**(config) # map alias SIP-First** **(config map alias SIP-First) # type firstLevel byRule** **(config map alias SIP-First) # rule add pass ipsrc 192.168.20.1 255.255.255.255 bidir** **(config map alias SIP-First) # rule add pass ipdst 192.168.20.1 255.255.255.255 bidir** **(config map alias SIP-First) # rule add pass ipsrc 192.168.20.128 255.255.255.255 bidir** **(config map alias SIP-First) # rule add pass ipdst 192.168.20.128 255.255.255.255 bidir** **(config map alias SIP-First) # to vport1** **(config map alias SIP-First) # from 10/3/x1,10/6/q1** **(config map alias SIP-First) # exit** **(config) #**
8.	Create a second level map for SIP whitelisting. Only one SIP whitelist map can be configure	**(config) # map alias sip-WL** **(config map alias sip-WL) # type secondLevel**

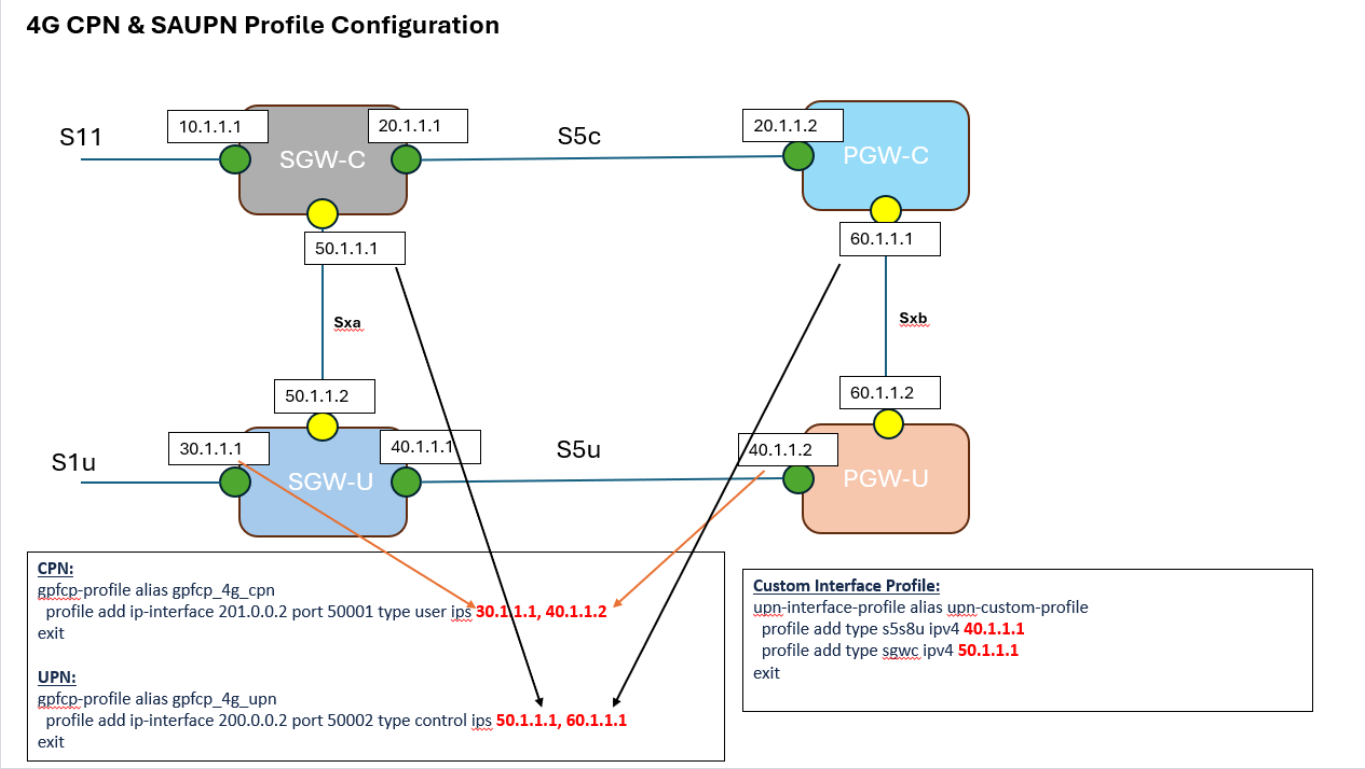
Step	Description	Command
		flowWhitelist-sip **(config map alias sip-WL) # use gsop sip-wl-lb** **(config map alias sip-WL) # to pg1** **(config map alias sip-WL) # from vport1** **(config map alias sip-WL) # exit** **(config) #**
9.	Create another second level map for SIP flow sampling. Up to five SIP flow sample maps can be configured.	**(config) # map alias map-sipFS** **(config map alias map-sipFS) # type secondLevel flowSample-sip** **(config map alias map-sipFS) # use gsop sip-fs-lb** **(config map alias map-sipFS) # flowsample add sip caller-id 408* percentage 50** **(config map alias map-sipFS) # flowsample add sip caller-id abc* percentage 75** **(config map alias map-sipFS) # to pg2** **(config map alias map-sipFS) # from vport1** **(config map alias map-sipFS) # exit** **(config) #**
10.	Add a shared collector for any uncorrelated RTP traffic from the virtual port and send it to a different tool port.	**(config) # map-scollector alias Collector** **(config map-scollector alias Collector) # from vport1** **(config map-scollector alias Collector) # collector 10/1/x26** **(config map-scollector alias Collector) # exit** **(config) #**
11.	Display the statistics for this example.	**(config) # show port stats**

Configure CPN-UPN Communication solution using CLI

CPN-UPN Communication enables traffic management based on Radio Access Network (RAN) and Network Slice attributes in the Standalone User Plane Node (UPN). This feature allows you to enable flow sampling and forward listing based on the RAN attributes and flow sampling based on the Network Slice attributes. This is achieved by establishing communication between GigaSMART 5G CUPS (CPN) and the Standalone UPN.

For more details regarding CPN-UPN communication topology, flow sample map rules, and forward list map rules, refer to CPN-UPN Communication for Support of RAN and Network Slice Attributes.

The below topology explains the 4G CPN & UPN Profile configuration.



The following configuration example explains the configuration flow for 4G CPN (when one 4G CPN communicates with one UPN).

Step	Description	Command
1.	Configure Ports.	**(config) # port 21/1/c1 mode 4x10G** **(config) # port 21/1/c1x1 type network** **(config) # port 21/1/c1x1 params admin enable** **(config) # port 21/1/c1x2 type network** **(config) # port 21/1/c1x2 params admin enable**
2.	Configure GigaStream.	**(config) # gigastream alias collector** **(config gigastream alias collector) # port-list 21/2/x21 params hash advanced** **(config gigastream alias collector) # exit**
3.	Configure one **gpfcp** profile for CPN and	**(config) # gpfcp-profile alias cpn-profile**

Step	Description	Command
	one **gpfcp** profile for UPN.	**(config gpfcp-profile alias cpn-profile) # profile add ip-interface 201.0.0.2 port 50001 type user ips 30.1.1.1, 40.1.1.2** **(config gpfcp-profile alias cpn-profile) # exit** **(config) # gpfcp-profile alias upn-profile** **(config gpfcp-profile alias upn-profile) # profile add ip-interface 200.0.0.2 port 50002 type control ips 50.1.1.1, 60.1.1.1** **(config gpfcp-profile alias upn-profile) # exit**
4.	Configure GigaSMART **UPN Interface profile**.	**(config) # upn-interface-profile alias upn-custom-profile** **(config upn-interface-profile alias upn-custom-profile) # profile add type s5s8u ipv4 40.1.1.1** **(config upn-interface-profile alias upn-custom-profile) # profile add type sgwc ipv4 50.1.1.1** **(config upn-interface-profile alias upn-custom-profile) # exit**
5.	Configure GigaSMART group.	**(config) # gsgroup alias gsgrp-cpn port-list 1/2/e1 hash advanced** **(config) # gsgroup alias gsgrp-upn port-list 1/4/e2 hash advanced**
6.	Configure IP interface.	**(config) # ip interface alias cpn-exporter** **(config ip interface alias cpn-exporter) # attach 1/3/c1x1** **(config ip interface alias cpn-exporter) # ip address 200.0.0.2 /24** **(config ip interface alias cpn-exporter) # gw 160.0.0.00** **(config ip interface alias cpn-exporter) # gsgroup add gsgrp-cpn** **(config ip interface alias cpn-exporter) # exit** **(config) # ip interface alias upn-exporter** **(config ip interface alias upn-exporter) # attach 1/3/c1x2** **(config ip interface alias upn-exporter) # ip address 201.0.0.2 /24** **(config ip interface alias upn-exporter) # gw 161.00.00.2** **(config ip interface alias upn-exporter) # gsgroup add gsgrp-upn** **(config ip interface alias upn-exporter) # exit**
7.	Configure Port Group.	**(config) # port-group alias pg-cpn** **(config port-group alias pg-cpn) # port-list 1/2/c3x1..c3x2** **(config port-group alias pg-cpn) # smart-lb enable** **(config port-group alias pg-cpn) # exit**

Step	Description	Command
		(config) # port-group alias pg-upn **(config port-group alias pg-upn) # port-list 1/3/c2x3..c2x4** **(config port-group alias pg-upn) # smart-lb enable** **(config port-group alias pg-upn) # exit**
8.	Configure GigaStream Host Exporter.	**(config) # apps exporter alias cpn-to-upn** **(config apps exporter alias cpn-to-upn) #type mobility-cups** **(config apps exporter alias cpn-to-upn) # source interface ip-interface cpn-exporter** **(config apps exporter alias cpn-to-upn) # source l4 port 50002** **(config apps exporter alias cpn-to-upn) # destination l4 protocol udp** **(config apps exporter alias cpn-to-upn) # destination l4 port 50001** **(config apps exporter alias cpn-to-upn) # destination l3 ip dscp 0** **(config apps exporter alias cpn-to-upn) # destination l3 ip ttl 64** **(config apps exporter alias cpn-to-upn) # destination l3 ip ver4 201.0.0.2** **(config apps exporter alias cpn-to-upn) # destination l3 protocol ipv4** **(config apps exporter alias cpn-to-upn) # gsgroup add gsgrp-cpn** **(config apps exporter alias cpn-to-upn) # exit** **(config) # apps exporter alias upn-to-cpn** **(config apps exporter alias upn-to-cpn) # type mobility-cups** **(config apps exporter alias upn-to-cpn) # source interface ip-interface upn-exporter** **(config apps exporter alias upn-to-cpn) # source l4 port 50001** **(config apps exporter alias upn-to-cpn) # destination l4 protocol udp** **(config apps exporter alias upn-to-cpn) # destination l4 port 50002** **(config apps exporter alias upn-to-cpn) # destination l3 ip dscp 0** **(config apps exporter alias upn-to-cpn) # destination l3 ip ttl 64** **(config apps exporter alias upn-to-cpn) # destination l3 ip ver4 200.0.0.2** **(config apps exporter alias upn-to-cpn) # destination l3 protocol ipv4** **(config apps exporter alias upn-to-cpn) # gsgroup add gsgrp-**

Step	Description	Command
		upn **(config apps exporter alias upn-to-cpn) # exit**
9.	Create a Forward List database.	**(config) # apps gtp-whitelist alias upn-wl-db create** **(config) # apps gtp-whitelist alias upn-wl-db add imsi 234567891234789** **(config) # apps gtp-whitelist alias upn-wl-db add ran 310.260.20942101** • **imsi—adds a single IMSI to a forward list.** • **ran—adds RAN value to add mcc.mnc.eci or mcc.mnc.nci value. The value must be given in the following format:** **NCI value in hexadecimal format and should add 0x as prefix.** **ECI value in decimal format and supports up to 9 digits.**
10.	Configure GigaSMART parameters.	**(config) # gsparams gsgroup gsgrp-cpn** **(config gsparams gsgroup gsgrp-cpn) # 3gpp-node-role control lte 1000** **(config gsparams gsgroup gsgrp-cpn) # gpfcp-delay interval 10** **(config gsparams gsgroup gsgrp-cpn) # gpfcp-profile add cpn-profile** **(config gsparams gsgroup gsgrp-cpn) # gtp-persistence enable** **(config gsparams gsgroup gsgrp-cpn) # exit** **(config) # gsparams gsgroup gsgrp-upn** **(config gsparams gsgroup gsgrp-upn) # 3gpp-node-role user 8000 stand-alone** **(config gsparams gsgroup gsgrp-upn) # gpfcp-delay interval 10** **(config gsparams gsgroup gsgrp-upn) # gpfcp-profile add upn-profile** **(config gsparams gsgroup gsgrp-upn) # gtp-persistence enable** **(config gsparams gsgroup gsgrp-upn) # gtp-whitelist add upn-wl-db** **(config gsparams gsgroup gsgrp-upn) # upn-interface-select type custom profile upn-custom-profile** **(config gsparams gsgroup gsgrp-upn) # exit**
11.	Configure GigaSMART operations.	**(config) # gsop alias fs-lb-cpn flow-ops gtp-flowsample lb app gtp metric hashing key imsi port-list gsgrp-cpn** **(config) # gsop alias fs-lb-upn flow-ops gtp-flowsample lb**

Step	Description	Command
		app gtp metric hashing key imsi port-list gsgrp-upn **(config) # gsop alias al-lb-upn flow-ops gtp-whitelist lb app gtp metric hashing key imsi port-list gsgrp-upn**
12	Configure Vport.	**(config) # vport alias vp-cpn gsgroup gsgrp-cpn** **(config) # vport alias vp-upn-ol gsgroup gsgrp-upn** **(config) # vport alias vp-upn-ol mode gtp-overlap**
13	Configure traffic map connection.	**(config) # map alias upn-pfcp** **(config map alias upn-pfcp) # type firstLevel byRule** **(config map alias upn-pfcp) # roles replace admin to owner_ roles** **(config map alias upn-pfcp) # rule add pass portdst 8805 bidir** **(config map alias upn-pfcp) # rule add pass ipfrag all-frag-no-first** **(config map alias upn-pfcp) # to vp-upn-ol** **(config map alias upn-pfcp) # from 1/4/c2** **(config map alias upn-pfcp) # exit** **(config) # map alias upn-gtpu** **(config map alias upn-gtpu) # type firstLevel byRule** **(config map alias upn-gtpu) # roles replace admin to owner_ roles** **(config map alias upn-gtpu) # rule add pass portdst 2152 bidir** **(config map alias upn-gtpu) # to vp-upn-ol** **(config map alias upn-gtpu) # from 1/2/c4x2** **(config map alias upn-gtpu) # exit** **(config) # map alias ControlPlane-cpn** **(config alias ControlPlane-cpn) # type firstLevel byRule** **(config alias ControlPlane-cpn) # roles replace admin to owner_roles** **(config alias ControlPlane-cpn) # rule add pass portsrc 2123 bidir** **(config alias ControlPlane-cpn) # to vp-cpn** **(config alias ControlPlane-cpn) # from 1/2/c4x1** **(config alias ControlPlane-cpn) # exit** **(config) # map alias FL-GPFCP-upn-to-cpn** **(config map alias FL-GPFCP-upn-to-cpn) # type firstLevel byRule**

Step	Description	Command
		(config map alias FL-GPFCP-upn-to-cpn) # roles replace admin to owner_roles (config map alias FL-GPFCP-upn-to-cpn) # rule add pass portdst 50001 (config map alias FL-GPFCP-upn-to-cpn) # to vp-cpn (config map alias FL-GPFCP-upn-to-cpn) # from 1/3/c1x1 (config map alias FL-GPFCP-upn-to-cpn) # exit (config) # map alias FL-GPFCP-cpn-to-UPN (config map alias FL-GPFCP-cpn-to-UPN) # type firstLevel byRule (config map alias FL-GPFCP-cpn-to-UPN) # roles replace admin to owner_roles (config map alias FL-GPFCP-cpn-to-UPN) # rule add pass portdst 50002 (config map alias FL-GPFCP-cpn-to-UPN) # to vp-upn-ol (config map alias FL-GPFCP-cpn-to-UPN) # from 1/3/c1x2 (config map alias FL-GPFCP-cpn-to-UPN) # exit (config) # map-scollector alias collector-cpn (config map-scollector alias collector-cpn) # roles replace admin to owner_roles (config map-scollector alias collector-cpn) # from vp-cpn (config map-scollector alias collector-cpn) # collector gtp-collector-cpn (config map-scollector alias collector-cpn) # exit (config) # map-scollector alias collector-upn (config map-scollector alias collector-upn) # roles replace admin to owner_roles (config map-scollector alias collector-upn) # from vp-upn-ol (config map-scollector alias collector-upn) # collector gtp-collector-upn (config map-scollector alias collector-upn) # exit (config) # map alias upn-fs1 (config map alias upn-fs1) # type secondLevel flowSample-ol (config map alias upn-fs1) # roles replace admin to owner_roles (config map alias upn-fs1) # use gsop fs-lb-upn (config map alias upn-fs1) # flowsample add gtp imsi

Step	Description	Command
		505024101215074 plmn-id 310.123 percentage 100 **(config map alias upn-fs1) # flowsample add gtp percentage 100** **(config map alias upn-fs1) # to pg-upn** **(config map alias upn-fs1) # from vp-upn-ol** **(config map alias upn-fs1) # exit** **(config) # map alias upn-wl1** **(config map alias upn-wl1) # type secondLevel flowWhitelist-ol** **(config map alias upn-wl1) # roles replace admin to owner_ roles** **(config map alias upn-wl1) # use gsop al-lb-upn** **(config map alias upn-wl1) # whitelist add gtp type all wl-alias upn-wl-db** **(config map alias upn-wl1) # whitelist add gtp interface N3** **(config map alias upn-wl1) # whitelist add gtp interface S1U** **(config map alias upn-wl1) # whitelist add gtp interface S5S8U** **(config map alias upn-wl1) # to pg-upn** **(config map alias upn-wl1) # from vp-upn-ol** **(config map alias upn-wl1) # exit** **(config) # map alias FlowSample-cpn** **(config map alias FlowSample-cpn) # type secondLevel flowSample** **(config map alias FlowSample-cpn) # roles replace admin to owner_roles** **(config map alias FlowSample-cpn) # use gsop fs-lb-cpn** **(config map alias FlowSample-cpn) # flowsample add gtp percentage 100** **(config map alias FlowSample-cpn) # to pg-cpn** **(config map alias FlowSample-cpn) # from vp-cpn** **(config map alias FlowSample-cpn) # exit**
14.	Configure Map-group.	**(config) # map-group alias upn-map-grp** **(config map-group alias upn-map-grp) # map-list upn-fs1,upn-wl1** **(config map-group alias upn-map-grp) # exit**

Related Show Commands

The following table summarizes the **show** commands related to the CPN-UPN communication configuration:

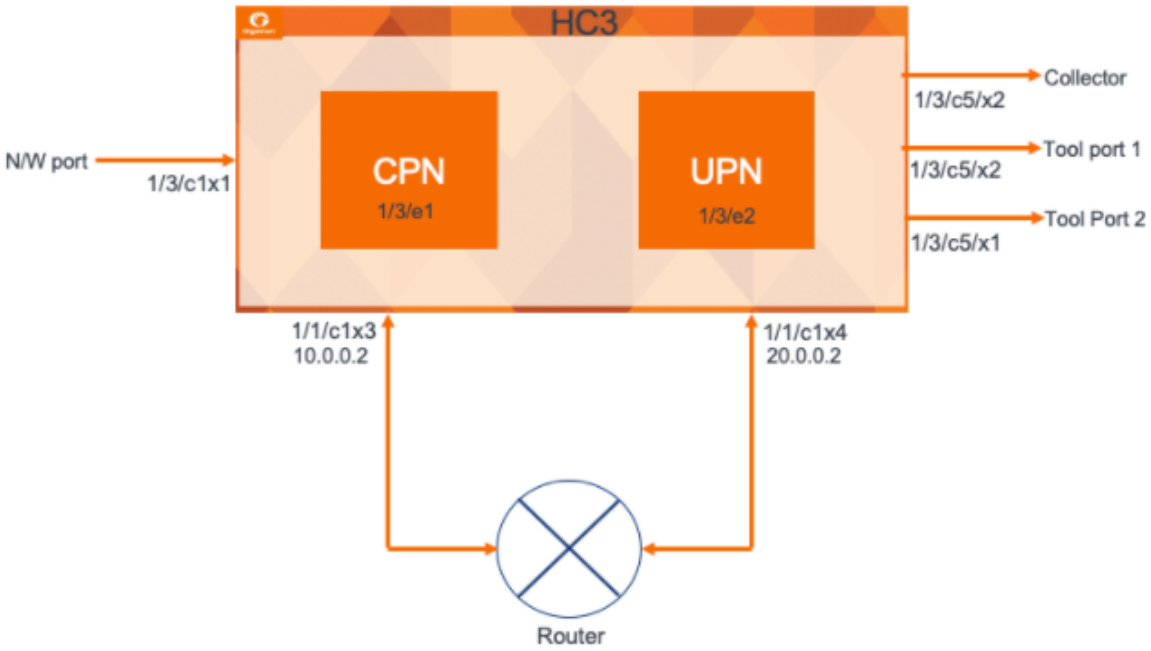
Task	Command
Displays a specified gpfcp profile.	**show gpfcp-profile alias <alias name>**
Displays all the created gpfcp profiles.	**show gpfcp-profile all**
Displays the Flow-ops report. Displays the Flow-ops report for 5G CPN.	**show gsgroup flow-ops-report alias <gsgroup alias> type flow-filtering any** **show gsgroup flow-ops-report alias <gsgroup alias> type flow-filtering any statistics** **show gsgroup flow-ops-report alias <gsgroup alias> type flow-5g any statistics**
Displays the Exporter statistics of a specific CPN or UPN.	**show apps exporter stats alias <exporter alias>**
Displays all the configured Exporters for CPN and UPN.	**show apps exporter stats all**

NOTE: For 5G CPN, the **gsparam eng-watchdog-timer** needs to be configured to a minimum value of 180 seconds.

Configuration of 5G Correlation

The 5G Correlation feature correlates the 5G Control and User packets to deliver it to different tool ports based on the filtering policies configured. The control and user packets are processed in Control Processing Node (CPN) and User Processing Node (UPN) located in same or different locations, and then the packets are sent to the tools.

This is a configuration example of 5G Correlation.



NOTE: It is recommended that you unconfigure and reload the GigaSMART card when configuring a particular GS group with 5G and switching back to 4G or vice versa.

S.No	Task	Command	Refer to Command
1.	Define port group/tool ports.	**(config) # gsgroup alias cpn-5g port-list 1/3/e1** **(config) # gsgroup alias upn-5g port-list 1/3/e2**	gsgroup
2.	Create a GSGROUP o CPN supports only single engine o UPN supports engine grouping	**(config) # gsparams gsgroup cpn-5g** **(config) # gsparams gsgroup upn-5g** **(config) #3gpp-node-role user**	gsgroup gsparams
3.	Define the node role.	**(config) # 3gpp-node-role control 5g 12000** **(config) #3gpp-node-role user**	gsparams
4.	Define the whitelist and attach it.		gsop
5.	Configure IP interface.	**(config) # ip interface alias cpn-ip**	

S.No	Task	Command	Refer to Command
		attach 1/1/c1x3 **ip address 10.0.0.2 /24** **gw 10.0.0.1** **gsgroup add cpn-5g** **exit** **ip interface alias upn-ip** **attach 1/1/c1x4** **ip address 20.0.0.2 /24** **gw 20.0.0.1** **gsgroup add upn-5g** **exit**	
6.	Create exporter/listener configs for SFFP communication.	**(config) # apps exporter alias cpn5000-5050exp** **type gtp-cups** **source interface ip-interface cpn-ip** **source l4 port 5000** **destination l4 protocol tcp** **destination l3 ip dscp 0** **destination l3 ip ttl 64** **destination l3 protocol ipv4** **gsgroup add cpn-5g** **exit** **(config) # apps listener alias upn-list-5050** **type gtp-cups** **l4 port-list 5050** **l4 protocol tcp** **l3 protocol ipv4** **l3 ttl 64** **l3 dscp 0** **mode l3 interface** **gsgroup add upn-5g** **exit**	apps exporter apps listener
7.	Create whitelist/ Flow sample GSOP (LB or non LB).		gsparams
8.	Create SFFP Profile and attach it to GSPARAMS.	**(config) # sffp-profile alias 5GCpnUpnSp**	sffp profile

S.No	Task	Command	Refer to Command
		profile add ip-interface 10.0.0.2 port 5000 type control sx-ips 10.80.2.30,fdc0:0:0:410::20 **profile add ip-interface 20.0.0.2 port 5050 type user sx-ips 10.50.2.70,fdc0:0:0:210::177** **exit**	
9.	Create a Vport.	**vport alias vport-cpn gsgroup cpn-5g** **vport alias vport-upn gsgroup upn-5g**	vport

S.No	Task	Command	Refer to Command
10.	Define first level map to send the traffic to Vport.	**map alias gtp-u** **type firstLevel byRule** **roles replace admin to owner_roles** **rule add pass portsrc 2152 bidir** **to vport-upn** **from 1/3/c1x1** **exit** **map alias 5g-ctl** **type firstLevel byRule** **roles replace admin to owner_roles** **rule add pass portsrc 80 bidir** **rule add pass portsrc 8080 bidir** **rule add pass portsrc 8090 bidir** **to vport-cpn** **from 1/3/c1x1** **exit** **map alias pfcp** **type firstLevel byRule** **roles replace admin to owner_roles** **rule add pass portsrc 8805 bidir** **to vport-cpn,vport-upn** **from 1/3/c1x1** **exit**	map
11.	Create second level map with Flow sample/ Whitelist rules.	**map alias cpn-5g-fs-map** **type secondLevel flowSample-5g** **roles replace admin to owner_roles** **use gsop fs-cpn** **flowsample add 5g percentage 100** **to 1/3/c5x1** **from vport-cpn** **exit** **map-scollector alias 5g-collector** **roles replace admin to owner_roles** **from vport-upn** **collector 1/3/c5x4** **exit**	map

S.No	Task	Command	Refer to Command
		map alias upn-5g-fs-map **type secondLevel flowSample-5g** **roles replace admin to owner_roles** **use gsop fs-upn** **flowsample add 5g percentage 100** **to 1/3/c5x2** **from vport-upn** **exit**	

GigaSMART FlowVUE

GigaSMART FlowVUE supports the following:

- flow-aware sampling of subscriber devices to filter and forward all flows sourced from a sampled set of subscriber device IPs
- flexible sampling on subscriber IPs and IP ranges, and at specified sampling rates
- user-configurable timeouts to detect and replace inactive devices
- IP-based sampling of flows and IP-based flows encapsulated in GTP-u tunnels

Sample of a Subset of Subscribers and Sample of all Subscribers Traffic

The following example show samples on GTP-u traffic where 10% of the subscribers are forwarded.

Step	Description	Command
1.	Configure one network and two tool type of ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias gsfvue flow-ops flow-sampling port-list gsg1**
4.	Configure sampling parameters,	**(config) # gsparams gsgroup gsg1 flow-sampling-device-ip-ranges add ip4addr 1.1.1.0 255.255.255.0** **(config) # gsparams gsgroup gsg1 flow-**

Step	Description	Command
		sampling-rate 10
5.	Create an ingress (first level) map. **Note**: In the rule, 2152 is GTP-u traffic.	**(config) # map alias to_tool** **(config map alias to_tool) # type regular byRule** **(config map alias to_tool) # to 1/1/x1** **(config map alias to_tool) # from 1/1/x3** **(config map alias to_tool) # rule add pass portsrc 2152** **(config map alias to_tool) # use gsop gsfvue** **(config map alias to_tool) # exit** **(config) #**

Sample a Subset of Subscribers and Sample a Subset of Traffic

FlowVUE can be used to reduce traffic to the monitoring tools. By combining FlowVUE with other GigaSMART applications such as APF, the traffic can be further reduced by filtering on specific Layer 4 application ports.

The following example samples on a subset of subscribers and forwards only the HTTP traffic related to the sampled subscriber set of devices.

Step	Description	Command
1.	Configure one network and two tool type of ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group. Also, configure APF.	**(config) # gsop alias gsfvue _apf flow-ops flow-sampling apf set port-list gsg1**
4.	Configure sampling parameters,	**(config) # gsparams gsgroup gsg1 flow-sampling-device-ip-ranges add ip4addr 1.1.1.0 255.255.255.0** **(config) # gsparams gsgroup gsg1 flow-sampling-rate 10**
5.	Configure virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**

Step	Description	Command
6.	Create a first level map and direct traffic to the virtual port. **Note**: In the rule, 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # rule add pass portsrc 2152** **(config map alias to_vp) # exit** **(config) #**
7.	Create a second level map and use the APF GigaSMART operation. APF performs filtering according to the gsrules, sending only matching traffic to the tool port.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # from vp1** **(config map alias map1) # gsrule add pass l4port dst pos 2 value 80** **(config map alias map1) # gsrule add pass l4port src pos 2 value 80** **(config map alias map1) # use gsop gsfvue_apf** **(config map alias map1) # exit** **(config) #**

GigaSMART Adaptive Packet Filtering (APF)

Adaptive Packet Filtering (APF) provides filtering on specific encapsulation protocol parameters. Additionally, it has the ability to look beyond the encapsulation protocol parameters into the original (encapsulated) data packet, to filter on source and destination IP or Layer 4 port numbers. APF offers the ability to look for content anywhere in the data packet and make intelligent filtering and forwarding decisions.

Adaptive Packet Filtering includes fragmentation awareness whereby all IP fragments associated with the filtered data packet are always forwarded allowing a complete view of the traffic stream for accurate analytics. APF also provides a powerful filtering engine that identifies content (based on patterns) across any part of the data packet, including the data packet payload.

APF Examples

The following are APF examples:

- Identify Social Security Numbers in User-Level Transactions
- Masking Social Security Numbers

- Filtering on Fiber Channel over Ethernet (FCOE) Traffic
- Multi-Encapsulation Filtering
- Filtering on Subscriber Device IP (User-Endpoint IP or UE-IP)
- Filtering on Inner Layer 2-4 Parameters for Unrecognized Headers
- GTP Tunnel ID-Based Filtering
- ERSPAN Tunneling
- Distributing Traffic Based on Inner IP Addresses and Inner TCP Port Values
- MPLS Label Based Filtering
- Combining APF with GigaSMART Operations
- Conditional Header Stripping
- Facilitating Overlapping Rules

Identify Social Security Numbers in User-Level Transactions

The following example looks for packets containing Social Security Numbers in an incoming traffic stream using pattern matching. Once a match is detected, the packets are forwarded to a monitoring tool for additional analysis.

Step	Description	Command
1.	Configure one network and two tool ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/1/e1**
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil apf set port-list gsgrp1**
4.	Create a virtual port.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create a first level map to forward traffic from network port 1/1/x3 to virtual port vp1.	**(config) # map alias map1** **(config map alias map1) # type firstLevel byRule** **(config map alias map1) # from 1/1/x3** **(config map alias map1) # to vp1** **(config map alias map1) # rule add pass ipver 4** **(config map alias map1) # exit** **(config) #**
6.	Create a second level map to forward traffic from the virtual port, vp,1 to GigaSMART with pattern matching.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1**

Step	Description	Command
		(config map alias map2) # use gsop gsfil **(config map alias map2) # to 1/1/x1** **(config map alias map2) # gsrule add pass pmatch RegEx "\d{3}-?\d{2}-?\d{4}" 40..80** **(config map alias map2) # exit** **(config) #**

Masking Social Security Numbers

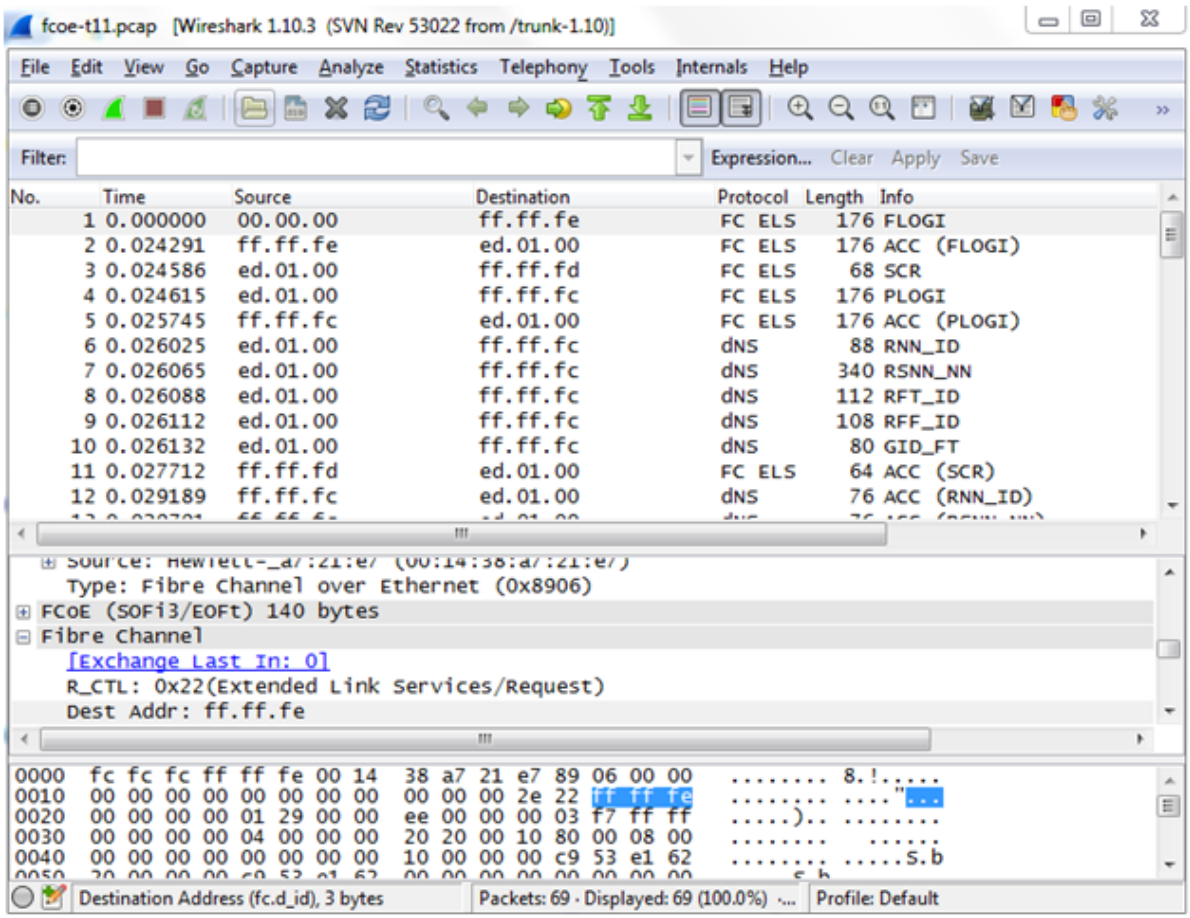
In the following pattern matching example, IPv4 packets contain Social Security Numbers (SSNs) in the format xxx-xx-xxxx. If the SSNs are between offset 40 and 80, they will be replaced with zeros.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1**
2.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias gsTraffic gsgroup gsgrp1**
3.	Create a first level map to direct traffic from network port 1/1/x1 to virtual port gsTraffic.	**(config) # map alias map1** **(config map alias map1) # type firstLevel byRule** **(config map alias map1) # from 1/1/x1** **(config map alias map1) # to gsTraffic** **(config map alias map1) # rule add pass ipver 4** **(config map alias map1) # exit** **(config) #**
4.	Configure the GigaSMART operation.	**(config) # gsop alias gsop1 apf set port-list gsgrp1**
5.	Create a second level map to direct traffic from the virtual port gsTraffic to GigaSMART.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from gsTraffic** **(config map alias map2) # use gsop gsop1** **(config map alias map2) # to 1/1/x6** **(config map alias map2) # gsrule add pass pmatch mask 0x00 RegEx "\d{3}-?\d{2}-?\d{4}" 40..80 (config map alias map2) # exit** **(config) #**

Step	Description	Command
6.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show vport** **(config) # show gsop** **(config) # show map**

Filtering on Fiber Channel over Ethernet (FCOE) Traffic

The flexibility offered by regular expression-based filters can be used as an infrastructure to classify traffic streams with protocol headers that are typically unsupported on traditional TAP/SPAN aggregation devices. In this example, regular expression-based filters are used for filtering on the source address in a Fiber Channel header.

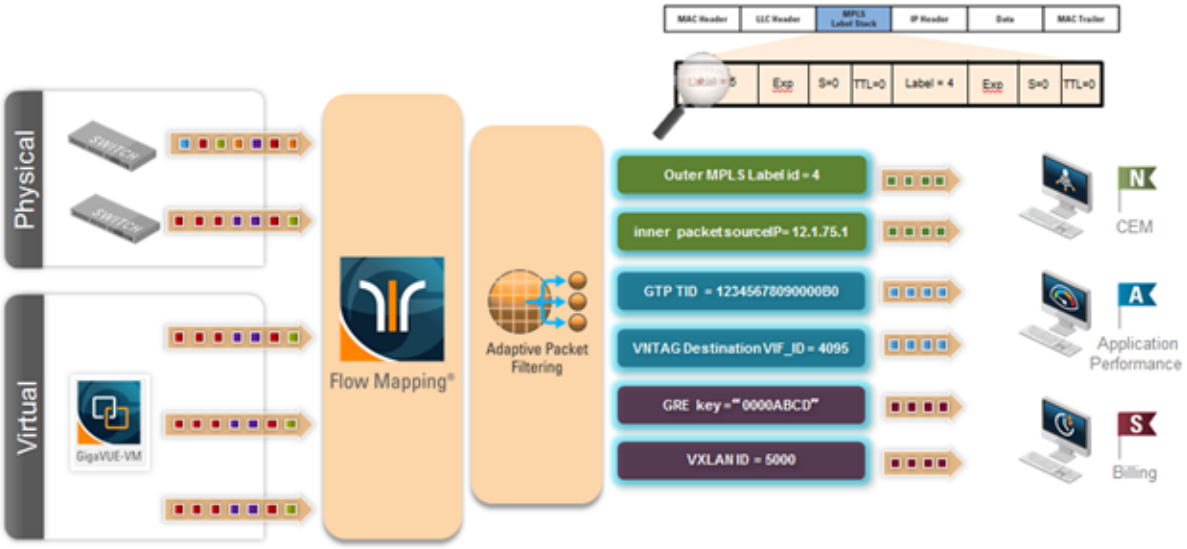


Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool**

Step	Description	Command
		(config) # port 1/1/x1 type tool
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward FCOE traffic to the virtual port.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass ethertype 8906** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on regular expression, using a string match to the destination address in the FCOE packet.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # gsrule add pass pmatch string "\xff\xff\xfe" 29** **(config map alias map1) # exit** **(config) #**

Multi-Encapsulation Filtering

In order to complement the mobility brought about by the virtualized server infrastructure, network virtualization overlays like VXLAN, VNTag, NVGRE are being designed and implemented in Data Centers and Enterprise environment. Across Service Provider environments, huge volumes of traffic are being tunneled over GTP. Until now, the Deep Observability Pipeline provided the option of stripping out these headers, thus providing visibility to monitoring tools that do not understand these overlays and encapsulation protocol. With APF, this capability is further enhanced where operators now have the option of making forwarding decisions based on the encapsulation and inner packet contents.



With encapsulation awareness enabled by APF, operators have multiple options to act on the packet including the flexibility to:

- Filter on encapsulation header parameters, Layer 2 – 4 parameters in the outer or inner headers (up to 5 layers of encapsulation) in any combination. For example:
  - Forward traffic specific to a subset of VXLAN ID's to one or more monitoring tools.
  - Distribute traffic based on MPLS label values across one or more monitoring tools.
- In combination with Header Stripping:
  - Implement "conditional" header-stripping, based on encapsulation header parameters or inner/outer packet contents, as follows:

    – Forward a subset of traffic "as-is" to monitoring tools that need these encapsulations for analysis.

    – Alternatively, strip out the outer headers/encapsulations and distribute traffic to monitoring tools that do not require these outer headers for analysis.
- Since APF is implemented as a second level map, operators can also implement overlapping rules where:
  - A copy of the traffic can be distributed across a group of monitoring tools.
  - A refined subset from the same incoming stream is distributed across a different set of tools.

Filtering on Subscriber Device IP (User-Endpoint IP or UE-IP)

Encapsulation awareness enabled by APF allows mobile operators to filter on Layer 2 – 4 header parameters found in an encapsulated packet.

This allows operators to filter and forward traffic specific to a mobile subscriber device or a group of subscriber devices, identified by their IP address (User-Endpoint IP) to one or more monitoring tools.

In this example, we are:

- Identifying and forwarding traffic from / to a UE-IP of 1.1.1.1 to a monitoring tool connected to 1/1/x1
- Identifying and forwarding traffic from / to a UE-IP of 1.1.1.2 to a different monitoring tool connected to tool port 1/1/x4

In many cases, the GTP control sessions are low-volume and are useful in providing some level of visibility in to the quality of experience of the subscribers. To this end, operators prefer to replicate the control sessions across all the monitoring tools, while filtering and forwarding a subset of the user-plane sessions to a subset of monitoring tools. The following example also illustrates configuration commands, leveraging the patented flow-mapping technology to replicate the GTP control sessions across all the monitoring tools involved in the traffic analysis.

Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward GTP-u traffic to the virtual port. **NOTE:** In the rule, 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass portsrc 2152** **(config map alias to_vp) # exit** **(config) #**
6.	Create a first level map to forward GTP-c traffic to the tools. **NOTE:** In the rule, 2123 is GTP-c	**(config) # map alias to_tool** **(config map alias to_tool) # type regular byRule** **(config map alias to_tool) # from 1/1/x3**

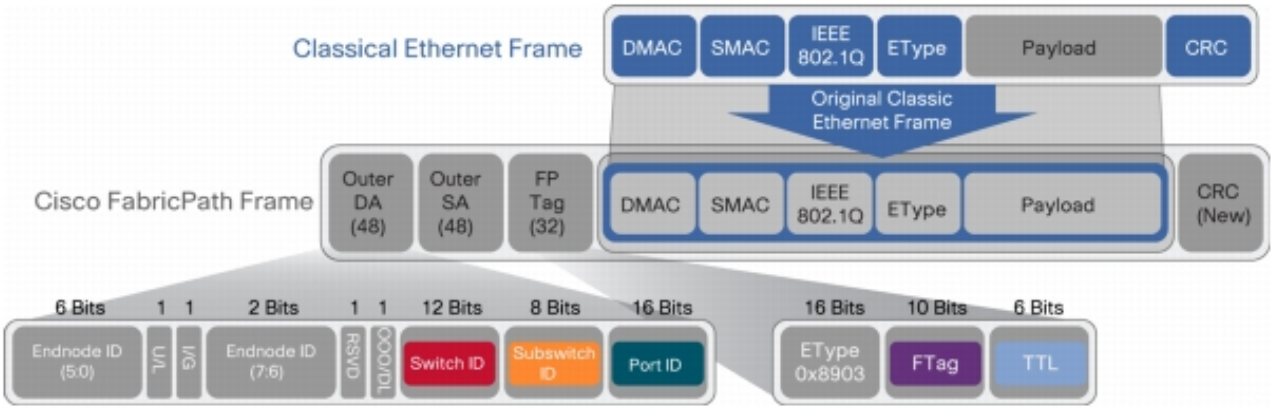
Step	Description	Command
	traffic.	**(config map alias to_tool) # to 1/1/x1,1/1/x4** **(config map alias to_tool) # rule add pass portsrc 2123** **(config map alias to_tool) # exit** **(config) #**
7.	Create a second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255** **(config map alias map1) # gsrule add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255** **(config map alias map1) # exit** **(config) #**
8.	Create another second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1** **(config map alias map2) # use gsop gsfil** **(config map alias map2) # to 1/1/x4** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # gsrule add pass ipv4 dst pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # exit** **(config) #**

Filtering on Inner Layer 2-4 Parameters for Unrecognized Headers

The flexibility of encapsulation awareness enables filtering on encapsulated contents even if APF does not recognize the outer encapsulation header. The following example illustrates a packet encapsulated in Fabric Path headers. Fabric Path headers (as shown in the figure) are mac-in-mac headers that are currently not recognized by APF. However operators can still filter and forward traffic flows based on Layer 2 – 4 parameters found in the encapsulated packets.

In this example, we are:

- Identifying and forwarding traffic from/to ip 1.1.1.1 in the inner / original packet to monitoring tool connected to tool port 1/1/x1
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the inner / original packet to monitoring tool connected to tool port 1/1/x4



Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward fabric path packets to the virtual port.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass ethertype 8903** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil**

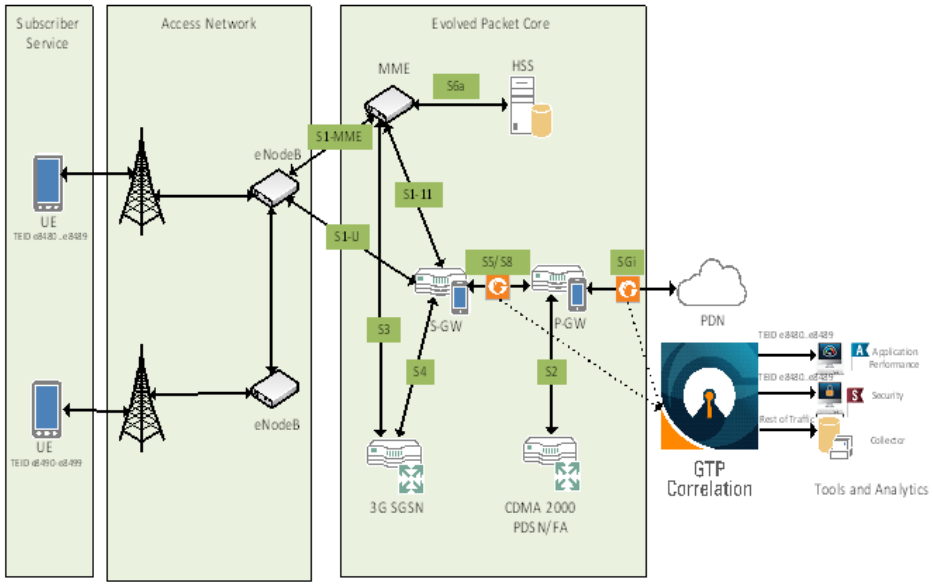
Step	Description	Command
		(config map alias map1) # to 1/1/x1 **(config map alias map1) # gsrule add pass ipv4 src pos 1 value 1.1.1.1 255.255.255.255** **(config map alias map1) # gsrule add pass ipv4 dst pos 1 value 1.1.1.1 255.255.255.255** **(config map alias map1) # exit** **(config) #**
7.	Create another second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1** **(config map alias map2) # use gsop gsfil** **(config map alias map2) # to 1/1/x4** **(config map alias map1) # gsrule add pass ipv4 src pos 1 value 1.1.1.2 255.255.255.255** **(config map alias map2) # gsrule add pass ipv4 dst pos 1 value 1.1.1.2 255.255.255.255** **(config map alias map2) # exit** **(config) #**

GTP Tunnel ID-Based Filtering

The following example demonstrates filtering and forwarding traffic based on tunnel IDs included as part of the GTP user-plane messages. It also illustrates the concept of a shared collector to which traffic not matching any of the configured filters can be optionally sent. GTP control sessions are forwarded to all the monitoring tools leveraging the power of Flow Mapping by filtering on Layer-4 UDP port 2123.

For GTP-u:

- Filter and forward teid ranges 0x001e8480..0x001e8489 to a monitoring tool
- Filter and forward teid ranges 0x001e8490..0x001e8499 to another monitoring tool
- Forward the rest of the traffic to a shared collector



Step	Description	Command
1.	Configure one network and three tool type of ports.	**(config) # port 1/3/x9 type network** **(config) # port 1/3/x15 type tool** **(config) # port 1/3/x13 type tool** **(config) # port 1/3/x14 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group. Packets processed by this operation are evaluated using Adaptive Packet Filtering (APF) rules.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map that directs GTP-u traffic from physical network port/s to the virtual port created in the previous step. **NOTE:** In the rule, 2152 is GTP-u traffic.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # from 1/3/x9** **(config map alias to_vp) # rule add pass portsrc 2152** **(config map alias to_vp) # exit** **(config) #**
6.	Create a first level map that directs	**(config) # map alias ctrl_to_tool**

Step	Description	Command
	GTP-u traffic from physical network port/s to the tool ports. **NOTE:** In the rule, 2123 is GTP-c traffic.	**(config map alias ctrl_to_tool) # type regular byRule** **(config map alias ctrl_to_tool) # to 1/3/x13,1/3/x15** **(config map alias ctrl_to_tool) # from 1/3/x9** **(config map alias ctrl_to_tool) # rule add pass portsrc 2123** **(config map alias ctrl_to_tool) # exit** **(config) #**
7.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, and matches tunnel IDs specified by the gsrule.	**(config) # map alias m1** **(config map alias m1) # type secondLevel byRule** **(config map alias m1) # use gsop gsfil** **(config map alias m1) # to 1/3/x15** **(config map alias m1) # from vp1** **(config map alias m1) # gsrule add pass gtp gtpu-teid range 0x001e8480..0x001e8489 subset none** **(config map alias m1) # exit** **(config) #**
8.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, and matches tunnel IDs specified by the gsrule.	**(config) # map alias m2** **(config map alias m2) # type secondLevel byRule** **(config map alias m2) # use gsop gsfil** **(config map alias m2) # to 1/3/x15** **(config map alias m2) # from vp1** **(config map alias m2) # gsrule add pass gtp gtpu-teid range 0x001e8490..0x001e8499 subset none** **(config map alias m2) # exit** **(config) #**
9.	Add a shared collector for any unmatched data and send it to the third tool port.	**(config) # map-scollector alias scoll** **(config map-scollector alias scoll) # from vp1** **(config map-scollector alias scoll) # collector 1/3/x14** **(config map-scollector alias scoll) # exit** **(config) #**

ERSPAN Tunneling

In this example, APF is used to filter packets based on ERSPAN ID. The ERSPAN header is not removed from the packet.

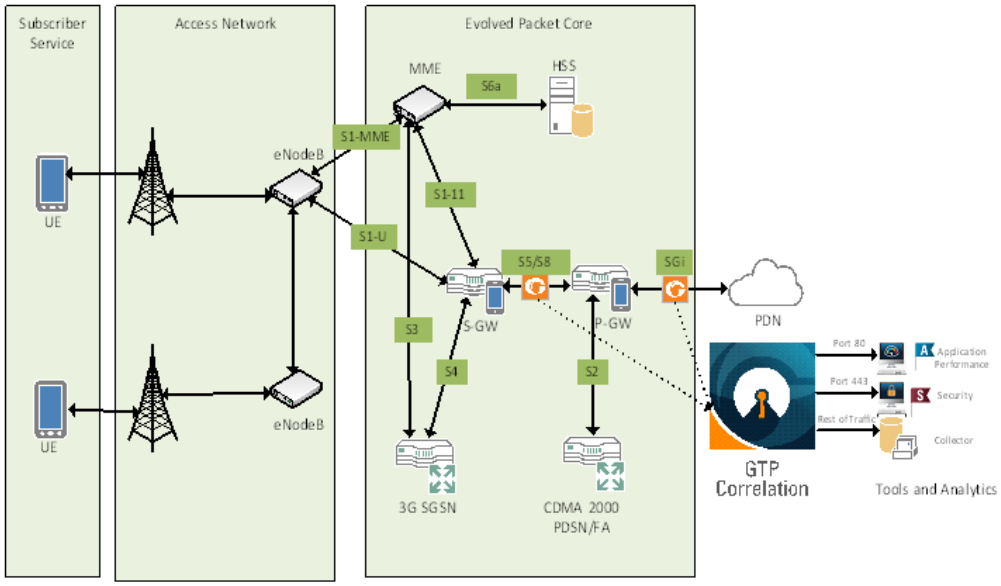
A second level map is configured in the example. A virtual port feeds traffic to the second level map. APF filters the packets and forwards those that match the filter criteria in the map.

Step	Description	Command
1.	Configure a tool type of port.	**(config) # port 1/1/g1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgp2 port-list 1/3/e2**
3.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp gsgroup gsgp2**
4.	Configure the GigaSMART operation and assign it to the GigaSMART group.	**(config) # gsop alias er2 apf set port-list gsgp2**
5.	Create a first level map.	**(config) # map alias test1a** **(config map alias test1a) # type firstLevel byRule** **(config map alias test1a) # to vp** **(config map alias test1a) # from 1/1/g3** **(config map alias test1a) # rule add pass macsrc 0000.0000.0000 0000.0000.0000** **(config map alias test1a) # exit** **(config) #**
6.	Create a second level map.	**(config) # map alias test1b** **(config map alias test1b) # type secondLevel byRule** **(config map alias test1b) # use gsop er2** **(config map alias test1b) # to 1/1/g1** **(config map alias test1b) # from vp** **(config map alias test1b) # gsrule add pass erspan id value 0** **(config map alias test1b) # exit** **(config) #**
7.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

Distributing Traffic Based on Inner IP Addresses and Inner TCP Port Values

In the following example, traffic is distributed based on inner IP addresses and inner TCP port values as follows:

- Packets from VLAN 20 with GTP inner IP 65.128.7.21 and 98.43.132.70, inner TCP port 80 is forwarded to one tool port
- Packets from VLAN 20 with GTP inner IP 65.128.7.21 and 98.43.132.70, inner TCP port 443 is forwarded to a second tool port
- All packets not matching these rules is forwarded to a third tool port



Step	Description	Command
1.	Configure one network and three tool type of ports.	**(config) # port 1/1/x1 type network** **(config) # port 1/1/x10 type tool** **(config) # port 1/1/x11 type tool** **(config) # port 1/1/x12 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsgrp1 port-list 1/1/e1**
3.	Configure the GigaSMART operation and assign it to the GigaSMART group. Packets processed by this operation are evaluated using Adaptive Packet Filtering (APF) rules.	**(config) # gsop alias g1 apf set port-list gsgrp1**
4.	Configure a virtual port and associate it with the GigaSMART group.	**(config) # vport alias gsTraffic gsgroup gsgrp1**
5.	Create a first level map that directs traffic from the physical network port/s to the virtual port created in the previous step.	**(config) # map alias map1** **(config map alias map1) # type firstLevel byRule** **(config map alias map1) # to gsTraffic**

Step	Description	Command
		(config map alias map1) # from 1/1/x1 **(config map alias map1) # rule add pass vlan 20 protocol udp portdst 2152** **(config map alias map1) # exit** **(config) #**
6.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches the rules, and sends the traffic to one tool port.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # use gsop g1** **(config map alias map2) # to 1/1/x10** **(config map alias map2) # from gsTraffic** **(config map alias map2) # gsrule add pass ipv4 dst pos 2 value 65.128.7.21 /32 ipv4 protocol pos 2 value tcp l4port dst pos 2 value 80** **(config map alias map2) # gsrule add pass ipv4 dst pos 2 value 98.43.132.70 /32 ipv4 protocol pos 2 value tcp l4port dst pos 2 value 80** **(config map alias map2) # exit** **(config) #**
7.	Create a second level map that takes traffic from the virtual port, applies the GigaSMART operation, matches the rules, and sends the traffic to another tool port.	**(config) # map alias map3** **(config map alias map3) # type secondLevel byRule** **(config map alias map3) # use gsop g1** **(config map alias map3) # to 1/1/x11** **(config map alias map3) # from gsTraffic** **(config map alias map3) # gsrule add pass ipv4 dst pos 2 value 65.128.7.21 /32 ipv4 protocol pos 2 value tcp l4port dst pos 2 value 443** **(config map alias map3) # gsrule add pass ipv4 dst pos 2 value 98.43.132.70 /32 ipv4 protocol pos 2 value tcp l4port dst pos 2 value 443** **(config map alias map3) # exit** **(config) #**
8.	Add a shared collector for any unmatched data and send it to the third tool port.	**(config) # map-scollector alias mapcl** **(config map-scollector alias mapcl) # from gsTraffic** **(config map-scollector alias mapcl) # collector 1/1/x12** **(config map-scollector alias mapcl) # exit** **(config) #**

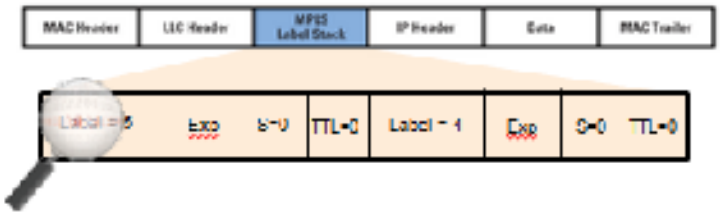
MPLS Label Based Filtering

Multiprotocol Label Switching (MPLS) is a mechanism in high-performance telecommunications networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (paths) between distant nodes rather than endpoints.

MPLS is a scalable, protocol-independent transport. In an MPLS network, data packets are assigned labels. Packet-forwarding decisions are made solely on the contents of this label, without the need to examine the packet itself. This allows one to create end-to-end circuits across any type of transport medium, using any protocol.

However in the context of Deep Observability Pipeline nodes, traffic flows encapsulated in MPLS labels cannot be filtered and forwarded. With the wide-scale adoption of MPLS as a technology across enterprise and service provider environments, the ability to classify traffic flows based on MPLS labels would be a huge value add to granularly control the flow of traffic to the monitoring tools. APF can be leveraged to filter and forward traffic flows based on MPLS label values. MPLS can stack multiple labels to form tunnels within tunnels. The flexibility of APF facilitates traffic classifications across up to 5 levels of MPLS label stacks in addition to the capability to filter and forward based on Layer 2-4 parameters found in the encapsulated packet. The following example illustrates filtering and forwarding traffic based on MPLS labels, as follows:

- Filter and forward traffic flows specific to mpls label = 4 at the second level in the MPLS label stack to tool 1
- Filter and forward traffic flows specific to mpls label = 3 at the first level in the MPLS label stack to tool 2



Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**

Step	Description	Command
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward traffic to the virtual port.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass ipver 4** **(config map alias to_vp) # rule add pass macsrc 00:00:00:00:00:00 bidir** **(config map alias to_vp) # exit** **(config) #**
6.	Create another second level map to filter on MPLS label.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # gsrule add pass mpls label pos 1 value 4** **(config map alias map1) # exit** **(config) #**
7.	Create another second level map to filter on MPLS label.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1** **(config map alias map2) # use gsop gsfil** **(config map alias map2) # to 1/1/x4** **(config map alias map1) # gsrule add pass mpls label pos 1 value 3** **(config map alias map2) # exit** **(config) #**

Combining APF with GigaSMART Operations

APF can also be combined with other GigaSMART functions including Header Stripping, Packet Slicing or Masking, De-duplication and FlowVUE. This provides network administrators and operators to perform a second layer of filtering in combination with the GigaSMART tool optimization and packet manipulation operations.

In the following example, operators can distribute traffic to monitoring tools based on de-capsulated contents, more specifically, after Header stripping VXLAN:

- Identifying and forwarding traffic from/to ip 1.1.1.1 from the decapsulated packets to monitoring tool connected to tool port 1/1/x1
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the decapsulated packets to monitoring tool connected to tool port 1/1/x4

NOTE: This can be applied to any protocol that is supported through header-stripping, for example:

- GTP, VXLAN, ISL, MPLS, MPLS+VLAN, VLAN, VN-Tag, fabric-path.
- This is also supported for Gigamon tunnel decapsulation.

Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operation.	**(config) # gsop alias gsfil_vxlanhs apf set strip-header vxlan 0 port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward VXLAN traffic to the virtual port. VXLAN accepts destination UDP ports 8472 and 4789. Starting in software version 4.5.01, VXLAN also accepts destination UDP port 48879.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass portsrc 8472** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil_vxlanhs** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255** **(config map alias map1) # gsrule add pass ipv4**

Step	Description	Command
		dst pos 2 value 1.1.1.1 255.255.255.255 **(config map alias map1) # exit** **(config) #**
7.	Create another second level map to filter on source and destination IP (bi-directional).	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1** **(config map alias map2) # use gsop gsfil_vxlanhs** **(config map alias map2) # to 1/1/x4** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # gsrule add pass ipv4 dst pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # exit** **(config) #**

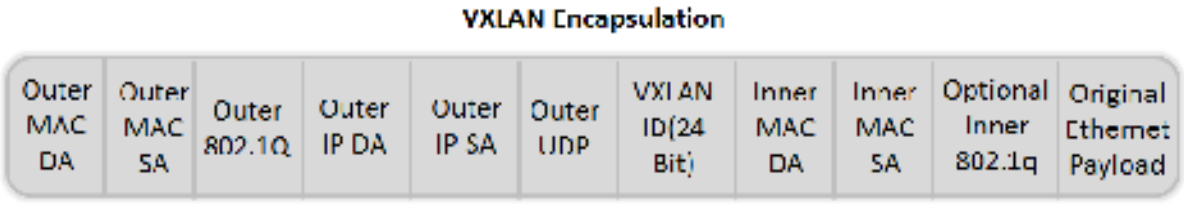
Conditional Header Stripping

Another use-case that can be addressed leveraging the flexibility of APF would be the capability to header strip packets based on specific contents found across the packet including the inner packet contents. Since the APF rules are enforced before any other GigaSMART operation, operators can filter based on encapsulation protocol values and /or encapsulated (original) packet contents and apply conditional Header Stripping operations.

The following example shows how an end-user can filter and strip out outer VXLAN headers for a subset of the traffic based on inner IP addresses, while sending the rest of the traffic “as-is” to monitoring tools that need the VXLAN headers for traffic analysis, as follows.

- Identifying and forwarding traffic from/to ip 1.1.1.1 in the inner / encapsulated packets to monitoring tool connected to tool port 1/1/x1 *after* Header Stripping VXLAN.
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the inner / encapsulated packets to monitoring tool connected to tool port 1/1/x4 *without* stripping the VXLAN header.

NOTE: This can be applied to any GigaSMART operation. While this example shows filtering based on inner packet contents, conditional SMART operations can be applied by filtering on encapsulation headers as well.



NOTE: This can be applied to any protocol that is supported through Header Stripping. GTP, VXLAN, ISL, MPLS, MPLS+VLAN, VLAN, VN-Tag, and fabric-path are all supported, as is Gigamon tunnel decapsulation.

Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operations.	**(config) # gsop alias gsfil_vxlanhs apf set strip-header vxlan 0 port-list gsg1** **(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward VXLAN traffic to the virtual port.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1** **(config map alias to_vp) # rule add pass portsrc 8472** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on source and destination IP (bi-directional), using first GigaSMART operation.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil_vxlanhs** **(config map alias map1) # to 1/1/x1** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255** **(config map alias map1) # gsrule add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255** **(config map alias map1) # exit** **(config) #**
7.	Create another second level map to filter on source and destination IP (bi-directional), using second GigaSMART operation.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1**

Step	Description	Command
		(config map alias map2) # use gsop gsfil **(config map alias map2) # to 1/1/x4** **(config map alias map1) # gsrule add pass ipv4 src pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # gsrule add pass ipv4 dst pos 2 value 1.1.1.2 255.255.255.255** **(config map alias map2) # exit** **(config) #**

Facilitating Overlapping Rules

Because APF is implemented as a second level map operation, APF can also be leveraged for implementing basic overlapping rules. For the same incoming input stream, a copy of the traffic can be sent out to a group of monitoring tools while a refined subset of the traffic stream can be sent to a different set of monitoring tools. Typically overlapping rules would be implemented by combining APF with the patented FlowMapping technology.

Note that Role-Based Access control in the case of APF is applied at the gsgroup / **e** port.

In the following example, for the same input stream:

- HTTP traffic is identified and distributed to a monitoring tool connected to tool port 1/1/x1.
- At the same time, the same stream of HTTP packets are being sent out after slicing unwanted packet contents to a different monitoring tool connected to tool port 1/1/x4.

Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**
3.	Configure the GigaSMART operations.	**(config) # gsop alias gsfil apf set port-list gsg1** **(config) # gsop alias gsfil_slice apf set slicing protocol none offset 150 port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward	**(config) # map alias to_vp**

Step	Description	Command
	traffic to the virtual port. Port 1/1/x1 and virtual port vp1 are sent destination port 80 traffic, which is HTTP.	**(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1,1/1/x1** **(config map alias to_vp) # rule add pass portdst 80 bidir** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on HTTP traffic and slice it.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil_slice** **(config map alias map1) # to 1/1/x4** **(config map alias map1) # gsrule add pass ipver pos 1 value 4** **(config map alias map1) # exit** **(config) #**
7.	Create another second level map for the rest of the traffic.	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # from vp1** **(config map alias map2) # use gsop gsfil** **(config map alias map2) # to 1/1/x1** **(config map alias map1) # gsrule add pass ipver pos 1 value 4** **(config map alias map2) # exit** **(config) #**

In the following example, for the same traffic stream, TCP traffic is sent to one monitoring tool while forwarding a subset of TCP flows specific to HTTP to another monitoring tool connected to tool port 1/1/x4.

Step	Description	Command
1.	Configure ports.	**(config) # port 1/1/x3 type network** **(config) # port 1/1/x4 type tool** **(config) # port 1/1/x1 type tool**
2.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg1 port-list 1/1/e1**

Step	Description	Command
3.	Configure the GigaSMART operations.	**(config) # gsop alias gsfil apf set port-list gsg1**
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsg1**
5.	Create a first level map to forward TCP traffic to the virtual port.	**(config) # map alias to_vp** **(config map alias to_vp) # type firstLevel byRule** **(config map alias to_vp) # from 1/1/x3** **(config map alias to_vp) # to vp1,1/1/x1** **(config map alias to_vp) # rule add pass protocol tcp** **(config map alias to_vp) # exit** **(config) #**
6.	Create a second level map to filter on HTTP traffic.	**(config) # map alias map1** **(config map alias map1) # type secondLevel byRule** **(config map alias map1) # from vp1** **(config map alias map1) # use gsop gsfil** **(config map alias map1) # to 1/1/x4** **(config map alias map1) # gsrule add pass l4port dst pos 2 value 80** **(config map alias map1) # gsrule add pass l4port src pos 2 value 80** **(config map alias map1) # exit** **(config) #**
7.	Display APF statistics	**(config) # show gsop stats alias forapf**

Gen 2 map and Gen 3 map configuration for Pattern Match RegEx

The following table highlights the differences between the Gen 2 and Gen 3 map configurations for Pattern Match RegEx.

Gen 2 map configuration	Gen 3 map configuration
map alias map1 type secondLevel byRule roles replace admin to owner_roles use gsop gsop2 gsrule add pass ipver any value 4	map alias map2 type secondLevel byRule roles replace admin to owner_roles use gsop gsop1 gsrule add pass ipver any value 4

Gen 2 map configuration	Gen 3 map configuration
ipv4 protocol any range 0x00..0xff subset none pmatch RegEx \\x53\\x54\\x3A **0..1750** gsrule add pass pmatch RegEx http:\\/\\/ (www\\.)etherealllll\\.com\\/distribution\\/rpms\\/ **0..1750** gsrule add pass pmatch RegEx www.digicert.commmm **0..1750** gsrule add drop ipver any value 4 ipv4 protocol any range 0x00..0xff subset none pmatch RegEx 20%match **0..1750** to 1/1/x6 from vp2 exit	ipv4 protocol any range 0x00..0xff subset none pmatch RegEx www.facebook.com **0** gsrule add drop pmatch RegEx \\x58\\x61\\x57\\x53\\x48\\x54 **0** gsrule add pass pmatch RegEx http:\\/\\/ (www\\.)ethereal\\.com\\/distribution\\/rpms\\/ **0** gsrule add pass pmatch RegEx www.digicert.com **0** gsrule add pass ipver any value 4 ipv4 protocol any range 0x00..0xff subset none pmatch RegEx trtrytryrtry **0** to 1/1/x6 from vp1 exit

GigaSMART Application Session Filtering (ASF) and Buffer ASF

Required Licenses: Adaptive Packet Filtering (APF) and Application Session Filtering (ASF)NOTE: The ASF license requires the APF license to be installed as a prerequisite.

Application Session Filtering (ASF) provides additional filtering on top of Adaptive Packet Filtering (APF). With APF, you can filter on any data patterns within a packet. With ASF, you apply the pattern matching and then send all the packet flows associated with the matched packet to monitoring or security tools.

ASF allows you to filter all traffic corresponding to a session. Use ASF to create a flow session and send the packets associated with the flow session to one or more tools. A flow session consists of one or more fields that you select to define the session. Either the packets for the whole session can be captured or only the packets following a pattern match.

ASF and Buffer ASF Examples

Refer to the following ASF examples (non-buffered):

- Example 1: ASF, Forward TCP Traffic
- Example 2: ASF, Forward VNC Traffic
- Example 3: ASF, Forward Traffic Matching a Pattern
- Example 4: ASF, Forward GTP Traffic

Refer to the following buffer ASF examples:

- Example 1: Buffer ASF, Drop Netflix Traffic
- Example 2: Buffer ASF, Drop YouTube Traffic
- Example 3: Buffer ASF, Drop Windows Update Traffic
- Example 4: Buffer ASF, Forward VNC Traffic
- Example 5: Buffer ASF, Forward HTTPS Traffic on Non-Standard Port

In addition to the examples in this document, the *Application Session Filtering Cookbook* provides a number of step-by-step recipes that show how to extract relevant flows with ASF, such as filtering YouTube traffic or emails with attachments. The cookbook also describes a methodology for identifying string patterns and regular expressions.

Example 1: ASF, Forward TCP Traffic

In Example 1, ASF is used with GigaSMART Load Balancing and Adaptive Packet Filtering to load balance TCP traffic among multiple tool ports. TCP SYN indicates the start of a connection. Once the TCP SYN packet is detected, subsequent packets belonging to the same TCP connection will be forwarded to a configured tool port. Packets belonging to the same connection will be sent to the same tool port, regardless of the number of connections.

In Example 1, the whole connection is captured because the first packet of the connection is captured by APF.

NOTE: This example uses APF to filter TCP packets to capture the SYN packet. Alternatively, use buffer ASF to capture a whole session by buffering packets.

The table illustrates the step-by-step ASF configuration commands for forwarding TCP traffic.

Step	Description	Command
1.	Create a flow session.	**(config) # apps asf alias asf4** **(config apps asf alias asf4) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf4) # exit** **(config) #**
2.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
3.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
4.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**

Step	Description	Command
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf asf4 lb app asf metric round-robin port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule captures the first packet of a session.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to portgrp1** **(config map alias map22) # gsrule add pass tcp ctl pos 1 value 2 mask none** **(config map alias map22) # exit** **(config) #**
9.	Display the configuration for this example.	**(config) # show port-group** **(config) # show gsgroup** **(config) # show gsop** **(config) # show gsop stats all** **Note**: In Gen3, the packets that matches the ruleSets are shown in **gsop stats all**. **Note**: In Gen2, the the packets that reach the v-port are shown in **gsop stats all**. **(config) # show load-balance port-group stats**

Example 2: ASF, Forward VNC Traffic

In Example 2, traffic from a Virtual Network Computing (VNC) application is forwarded from network port 1/1/x1 to tool port 1/1/x6. Packets will be matched with a VNC signature. Once a packet is matched, subsequent packets with the same IPv4 5tuple will be forwarded to the same destination as the matching packet. By default, both the forward and the reverse traffic of the same session will be captured and forwarded.

The table illustrates the step-by-step ASF configuration commands for forwarding VNC traffic.

Step	Description	Command
1.	Create a flow session.	**(config) # apps asf alias asf1** **(config apps asf alias asf1) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf1) # exit** **(config) #**
2.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
3.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf asf1 port-list gsgrp1
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
6.	Create a second level egress map. The gsrule contains the VNC signature.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add pass pmatch RegEx "^rfb 00[1-9]\.00[0-9]\x0a$" 16..1000** **(config map alias map22) # exit** **(config) #**
7.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

Example 3: ASF, Forward Traffic Matching a Pattern

In Example 3, the traffic that matches a particular pattern (ymsg|ypns|yhoo) is forwarded from network port 1/1/x1 to tool port 1/1/x6 after adding a VLAN tag. Packets will be matched with the special signature. Once a packet is matched, subsequent packets with the same source IP, source port, and VLAN ID will be forwarded to the same destination as the matching packet (after the VLAN header is inserted). By default, both the forward and the reverse traffic of the same session will be captured and forwarded.

The table illustrates the ASF configuration commands for forwarding pattern-matched traffic with VLAN tagging.

Step	Description	Command
1.	Create a flow session and other parameters.	**(config) # apps asf alias asf2** **(config apps asf alias asf2) # sess-field add ipv4-src outer** **(config apps asf alias asf2) # sess-field add l4port-src outer** **(config apps asf alias asf2) # sess-field add vlan-id pos 1** **(config apps asf alias asf2) # packet-count 50** **(config apps asf alias asf2) # exit** **(config) #**
2.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
3.	Configure the GigaSMART operation.	**(config) #** gsop alias gsop1 apf set add-header vlan 1000 **asf asf2** port-list gsgrp1
4.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
5.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
6.	Create a second level map. The gsrule contains the special signature.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1**

Step	Description	Command
		(config map alias map22) # use gsop gsop1 **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add pass pmatch RegEx "(ymsg|ypns|yhoo)" 16..1000** **(config map alias map22) # exit** **(config) #**
7.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

Example 4: ASF, Forward GTP Traffic

In Example 4, GTP traffic from network port 1/1/x1 is load balanced based on inner IP and tunnel ID to four tool ports: 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4. APF filters GTP-u packets. Once a packet is matched, subsequent packets in the same direction with the same gtpu-teid and inner IP will be forwarded to the same destination as the matching packet. In Example 4, both the outer and inner IP are IPv4.

Step	Description	Command
1.	Create a flow session and other parameters.	**(config) # apps asf alias asf3** **(config apps asf alias asf3) # sess-field add gtpu-teid** **(config apps asf alias asf3) # sess-field add ipv4 inner** **(config apps asf alias asf3) # bi-directional disable** **(config apps asf alias asf3) # timeout 90** **(config apps asf alias asf3) # exit** **(config) #**
2.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
3.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
4.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf3** lb app asf metric lt-conn port-list gsgrp1

Step	Description	Command
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass protocol udp portdst 2152** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to portgrp1** **(config map alias map22) # gsrule add pass ipv4 protocol pos 1 value udp l4port dst pos 1 value 2152** **(config map alias map22) # exit** **(config) #**
9.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show port-group** **(config) # show map** **(config) # show apps asf**

Example 1: Buffer ASF, Drop Netflix Traffic

In Example 1, the goal is to drop all Netflix traffic. The flow session is defined by the 5tuple field and the first occurrence of VLAN ID. The Netflix traffic is expected to be identified in the first 6 packets of a session. (Configure the maximum number of packets buffered before the match to 5.) A maximum of 3 million sessions is specified.

The table illustrates the Buffer ASF configuration commands to identify and drop Netflix traffic.

Step	Description	Command
1.	Configure a GigaSMART group	**(config) # gsgroup alias gsgrp1 port-list**

Step	Description	Command
	and associate it with GigaSMART engine ports.	**1/3/e1,1/3/e2**
2.	Define the maximum number of sessions, in millions.	**(config) # gsparams gsgroup gsgrp1 resource buffer-asf 3**
3.	If needed, reload the GigaSMART line card or module to allocate the resources for buffer ASF.	**(config) # card slot 3 down** Then to bring the GigaSMART line card or module back up: **(config) # no card slot 3 down**
4.	Create a flow session, specify the buffer count before the match, and enable buffering. **Note**: The default protocol is TCP, so it does not need to be specified.	**(config) # apps asf alias asf2** **(config apps asf alias asf2) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf2) # sess-field add vlan-id pos 1** **(config apps asf alias asf2) # buffer-count-before-match 5** **(config apps asf alias asf2) # buffer enable** **(config apps asf alias asf2) # exit** **(config) #**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf2** port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule specifies the traffic to drop, using keywords. Buffered packets and all subsequent packets will be dropped.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add drop pmatch protocol tcp pos 1 RegEx “netflix|nflxvideo|nflximg|Netflix|nflxext” 0..1460** **(config map alias map22) # exit**

Step	Description	Command
		(config) #
9.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsparams** **(config) # show apps asf** **(config) # show gsop** **(config) # show vport** **(config) # show map**

Example 2: Buffer ASF, Drop YouTube Traffic

In Example 2, the goal is to drop all YouTube traffic. The YouTube traffic is expected to be identified in the first 7 packets of a session. (Configure the maximum number of packets buffered before the match to 6.) A maximum of 4 million sessions is specified.

The following table illustrates the Buffer ASF configuration commands to detect and drop YouTube traffic.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
2.	Define the maximum number of sessions, in millions.	**(config) # gsparams gsgroup gsgrp1 resource buffer-asf 4**
3.	If needed, reload the GigaSMART line card or module to allocate the resources for buffer ASF.	**(config) # card slot 3 down** Then to bring the GigaSMART line card or module back up: **(config) # no card slot 3 down**
4.	Create a flow session, specify the buffer count before the match, and enable buffering. **Note**: The default protocol is TCP, so it does not need to be specified.	**(config) # apps asf alias asf2** **(config apps asf alias asf2) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf2) # buffer-count-before-match 6** **(config apps asf alias asf2) # buffer enable** **(config apps asf alias asf2) # exit** **(config) #**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf2** port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART	**(config) # vport alias vp1 gsgroup gsgrp1**

Step	Description	Command
	group.	
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule specifies the traffic to drop, using keywords. Buffered packets and all subsequent packets will be dropped.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add drop pmatch protocol tcp pos 1 RegEx “youtube|ytimg|yt3.ggpht|tubeMogul|tmogul” 0..1460** **(config map alias map22) # exit** **(config) #**
9.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsparams** **(config) # show apps asf** **(config) # show gsop** **(config) # show vport** **(config) # show map**

Example 3: Buffer ASF, Drop Windows Update Traffic

In Example 3, the goal is to drop all Windows update traffic. The Windows update traffic is expected to be identified on the HTTP request packet of a session. A maximum of 2 million sessions is specified.

The following table illustrates the Buffer ASF configuration commands to detect and drop Windows Update traffic.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with GigaSMART	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**

Step	Description	Command
	engine ports.	
2.	Define the maximum number of sessions, in millions.	**(config) # gsparams gsgroup gsgrp1 resource buffer-asf 2**
3.	If needed, reload the GigaSMART line card or module to allocate the resources for buffer ASF.	**(config) # card slot 3 down** Then to bring the GigaSMART line card or module back up: **(config) # no card slot 3 down**
4.	Create a flow session, specify the buffer count before the match, and enable buffering. **Note**: The default protocol is TCP, so it does not need to be specified.	**(config) # apps asf alias asf2** **(config apps asf alias asf2) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf2) # buffer-count-before-match 3** **(config apps asf alias asf2) # buffer enable** **(config apps asf alias asf2) # exit** **(config) #**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf2** port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule specifies the traffic to drop. Buffered packets and all subsequent packets will be dropped.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add drop pmatch protocol tcp pos 1 RegEx “msdownload/update/software” 0..1460** **(config map alias map22) # exit** **(config) #**

Step	Description	Command
9.	Display the configuration for this example.	**(config) # show gsparams** **(config) # show apps asf** **(config) # show map**

Example 4: Buffer ASF, Forward VNC Traffic

In Example 4, the goal is to forward VNC traffic from network port 1/1/x1 to tool port 1/1/x6. All packets belonging to the TCP connection need to be sent to the tool port. The first data packet after the TCP handshake is expected to contain the VNC pattern match. A maximum of 2 million sessions is specified.

The following table illustrates the Buffer ASF configuration commands to detect and forward VNC traffic.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
2.	Define the maximum number of sessions, in millions.	**(config) # gsparams gsgroup gsgrp1 resource buffer-asf 2**
3.	If needed, reload the GigaSMART line card or module to allocate the resources for buffer ASF.	**(config) # card slot 3 down** Then to bring the GigaSMART line card or module back up: **(config) # no card slot 3 down**
4.	Create a flow session, specify the buffer count before the match, and enable buffering. **Note**: The default protocol is TCP, so it does not need to be specified.	**(config) # apps asf alias asf1** **(config apps asf alias asf1) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf1) # buffer-count-before-match 3** **(config apps asf alias asf1) # buffer enable** **(config apps asf alias asf1) # exit** **(config) #**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf1** port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule**

Step	Description	Command
		(config map alias map11) # from 1/1/x1 **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule specifies the traffic to pass. Buffered packets and all subsequent packets will be passed.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add pass pmatch protocol tcp pos 1 RegEx "^rfb 00[1-9]\.00[0-9]\x0a$" 0** **(config map alias map22) # exit** **(config) #**
9.	Display the configuration for this example.	**(config) # show gsparams** **(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

Example 5: Buffer ASF, Forward HTTPS Traffic on Non-Standard Port

In Example 5, the goal is to forward HTTPS traffic that uses a non-standard Layer 4 port. All packets belonging to the TCP connection need to be sent to the tool port. A maximum of 5 million sessions is specified.

The following table illustrates the Buffer ASF configuration commands to forward HTTPS traffic on non-standard ports.

Step	Description	Command
1.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
2.	Define the maximum number of sessions, in millions.	**(config) # gsparams gsgroup gsgrp1 resource buffer-asf 5**
3.	If needed, reload the GigaSMART line card or module to allocate the resources for buffer ASF.	**(config) # card slot 3 down** Then to bring the GigaSMART line card or module back up:

Step	Description	Command
		(config) # no card slot 3 down
4.	Create a flow session, specify the buffer count before the match, and enable buffering. **Note**: The default protocol is TCP, so it does not need to be specified.	**(config) # apps asf alias asf2** **(config apps asf alias asf2) # sess-field add ipv4-5tuple outer** **(config apps asf alias asf2) # buffer-count-before-match 3** **(config apps asf alias asf2) # buffer enable** **(config apps asf alias asf2) # exit** **(config) #**
5.	Configure the combined GigaSMART operation.	**(config) #** gsop alias gsop1 apf set asf **asf2** port-list gsgrp1
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create a first level map.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass ipver 4** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map. The gsrule specifies the traffic to pass. The RegEx expression identifies the traffic as SSL. Buffered packets and all subsequent packets will be passed.	**(config) # map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to 1/1/x6** **(config map alias map22) # gsrule add pass pmatch protocol tcp pos 1 RegEx "\x16\x03.{3}\x01" 0** **(config map alias map22) # exit** **(config) #**
9.	Display the configuration for this example.	**(config) # show gsgroup** **(config) # show gsop** **(config) # show map**

GigaSMART NetFlow Generation

NetFlow Generation is a simple and effective way to increase visibility into traffic flows and usage patterns across systems. The flow-generated data can be used to build relationships and usage patterns between nodes on the network. Routers and switches that support NetFlow can collect IP traffic statistics to be exported as NetFlow records.

However, the processor and memory load of enabling NetFlow can cause service degradation and affect their ability to pass traffic without introducing latency and packet drops. Due to this processing overhead, sampled NetFlow is implemented in most of the high-end routers. Sampling in every "N" packets for NetFlow processing can severely limit the visibility needed to monitor flows.

Configure NetFlow Generation Examples

The following sections provide examples of NetFlow Generation. Refer to the following:

- Example 1: NetFlow Generation Configuration
- Example 2: NetFlow Generation Configuration
- Example 3: NetFlow Generation Configuration
- Example 4: NetFlow Generation Configuration

For details on the CLI commands in the following sections, refer to the following commands in the reference section:

- apps netflow
- gsgroup
- gsop
- gsparams
- map
- port
- ip interface

Example 1: NetFlow Generation Configuration

In Example 1, the steps set up a typical NetFlow Generation configuration.

Ex 1, Step 1: Configure the Exporter

Configure one or more NetFlow Generation Exporter(s). There can be up to 6 NetFlow Generation Exporters for each NetFlow Generation Monitor.

The following command options show the configuration of collector parameters for the NetFlow records that are exported.

The following command descriptions apply:

- **format netflow version** - The version is ipfix, netflow-v5, or netflow-v9.

  NOTE: The NetFlow version must be configured with the same version of the Exporter and the Record. If no version is specified, version 9 is the default.

- **destination ip4addr** - The IP address of the NetFlow/IPFIX collector.
- **transport** - The UDP port of the collector.
- **ttl** - The Time to Live of the packet.
- **dscp** - The DSCP priority of the packet.
- **template-refresh-interval** - After each template-refresh-interval, the record template is sent to the collector. Also, the option template is sent.
- **snmp enable** - Enables SNMP packet support for the NetFlow exporter.

Step	Description	Command
1.	Configure the exporter. The exporter (exp4) will be used in Ex 1, Step 7: Configure Exporter Associated to IP Interface Tool Port.	**(config) # apps netflow exporter alias exp4** **(config apps netflow exporter alias exp4) # format netflow version ipfix** **(config apps netflow exporter alias exp4) # destination ip4addr 20.20.20.20** **(config apps netflow exporter alias exp4) # transport udp 2055** **(config apps netflow exporter alias exp4) # ttl 64** **(config apps netflow exporter alias exp4) # dscp 10** **(config apps netflow exporter alias exp4) # template-refresh-interval 60** **(config apps netflow exporter alias exp4) # snmp enable** **(config apps netflow exporter alias exp4) # exit** **(config) #**
2.	Display the exporter configuration.	**(config) # show apps netflow exporter**

Ex 1, Step 2: Configure the Record

Configure one or more NetFlow Generation Records, which have the following:

- **match** parameters that identify unique flows
- **collect** parameters that identify fields you want to collect for the unique flows

NOTE: NetFlow v9 and IPFIX let you configure Match/Key and Collect/Non-Key elements. For details refer to *NetFlow Generation Match/Key and Collect/Non-Key Elements* on page 698.

The following table describes the commands for NetFlow Generation Records:

Parameter	Description
netflow-version	The version is either netflow-v9 or ipfix. **NOTE:** The NetFlow version must be configured with the same version as the Exporter and the Record. If no version is specified, version 9 is the default.
export-blank-pen	The parameter that exports a record containing both private and non-private enterprise elements when during runtime, the private enterprise element is empty.
exporter	The parameter that assigns an exporter to a NetFlow record.
match fields	The parameters that identify unique flows. The available Match/Key fields are based on the configured NetFlow version.
collect fields	The parameters that identify what you want to collect for the unique flows. The number of Collect/Non-Key elements in a record can be up to 32.
sampling	The parameter that configures the sampling rate and enables sampling.

In this example, the IP source and destination address on the incoming traffic is used to identify network traffic between the unique pair of source and destination addresses. Once unique flows are identified, the following parameters are collected and exported for each flow:

- IP source and destination address
- Total number of packets and bytes received that match the unique flows
- IPv4 protocol
- Transport source and destination ports
- Input and output interface, plus interface name
- Packet URL
- DNS response name
- Timestamp for the beginning and end of flow

In this example, the collect fields are in two records. Both records must be added to the monitor.

NOTE: Configure the NetFlow version prior to configuring the match and collect parameters because the subsequent parameters depend on the netflow-version configured. If no version is specified, the version 9 is the default (netflow-v9).

Step	Description	Command
1.	Configure a record. The NetFlow version must be the same as the NetFlow version specified in Ex 1, Step 1: Configure the Exporter. The record (rec2) will be used in Ex 1, Step 3: Configure the Monitor.	**(config) # apps netflow record alias rec2** **(config apps netflow record alias rec2) # netflow-version ipfix** **(config apps netflow record alias rec2) # match add ipv4 source address** **(config apps netflow record alias rec2) # match add ipv4 destination address** **(config apps netflow record alias rec2) # collect add ipv4 source address** **(config apps netflow record alias rec2) # collect add ipv4 destination address** **(config apps netflow record alias rec2) # collect add counter packets** **(config apps netflow record alias rec2) # collect add counter bytes** **(config apps netflow record alias rec2) # collect add ipv4 protocol** **(config apps netflow record alias rec2) # collect add transport source-port** **(config apps netflow record alias rec2) # collect add interface input physical** **(config apps netflow record alias rec2) # collect add interface input name** **(config apps netflow record alias rec2) # collect add transport destination-port** **(config apps netflow record alias rec2) # collect add interface output physical** **(config apps netflow record alias rec2) # exporter add exp4** **(config apps netflow record alias rec2) # sampling set 1 in 10** **(config apps netflow record alias rec2) # exit** **(config) #**
2.	Configure a second record. The NetFlow version must be the same as the NetFlow version specified in Ex 1, Step 1: Configure the Exporter. The match fields must be the same as in Step 1. Each record must have the same match fields but differing collect fields.	**(config) # apps netflow record alias rec3** **(config apps netflow record alias rec3) # netflow-version ipfix** **(config apps netflow record alias rec3) # match add ipv4 source address** **(config apps netflow record alias rec3) # match add ipv4 destination address** **(config apps netflow record alias rec3) # collect add private pen gigamon http url**

Step	Description	Command
	The record (rec3) will be used in Ex 1, Step 3: Configure the Monitor.	**(config apps netflow record alias rec3) # collect add private pen gigamon dns query-name** **(config apps netflow record alias rec3) # collect add private pen gigamon dns response-name number-of-collects 2** **(config apps netflow record alias rec3) # collect add timestamp sys-uptime first** **(config apps netflow record alias rec3) # collect add timestamp sys-uptime last** **(config apps netflow record alias rec2) # exporter add exp4** **(config apps netflow record alias rec2) # sampling set 1 in 20** **(config apps netflow record alias rec3) # exit** **(config) #**
3.	Display the record configuration.	**(config) # show apps netflow record**

Ex 1, Step 3: Configure the Monitor

Configure a NetFlow Generation Monitor and associate the NetFlow Generation Record to the specified NetFlow Generation Monitor.

The following commands show the binding of the records. The commands also define the cache (holding statistics for unique flows).

The following command descriptions apply:

- **record add** - Records generated for the flow are defined in the record and are stored in the internal cache.
- **cache timeout event transaction-end** - Applies to the TCP flow. The flow is "flushed out" to the Exporter after detecting a FIN or RST.
- **cache timeout inactive** - Inactive flows are "flushed out" to the Exporter after this timeout, which is set in seconds.
- **cache timeout active** - Despite the flow being active, it is "flushed out" to the Exporter after this timeout, which is set in seconds.
- **sampling** - Enables or disables single-rate sampling and defines the sampling rate by specifying a number for 1 in N, where N is the packet count from 10 to 16000.

Step	Description	Command
1.	Configure the monitor. The	**(config) # apps netflow monitor alias mon2**

Step	Description	Command
	monitor (mon2) will be used in Ex 1, Step 8: Configure GigaSMART Params to Add a Monitor. The records (rec2 and rec3) were created in Ex 1, Step 2: Configure the Record. In this example, NetFlow sampling is enabled. The sampling rate is 1 in 1024.	**(config apps netflow monitor alias mon2) # record add rec2(** **config apps netflow monitor alias mon2) # record add rec3** **(config apps netflow monitor alias mon2) # cache timeout event transaction-end** **(config apps netflow monitor alias mon2) # cache timeout inactive 15** **(config apps netflow monitor alias mon2) # cache timeout active 60** **(config apps netflow monitor alias mon2) # sampling set single-rate** **(config apps netflow monitor alias mon2) # sampling single-rate 1 in 1024** **(config apps netflow monitor alias mon2) # exit** **(config) #**
2.	Display the monitor configuration.	**(config) # show apps netflow monitor**

Ex 1, Step 4: Configure the gsgroup

> **NOTE:** The GigaSMART group can contain multiple GigaSMART engine ports.

Configure a GigaSMART group and associate it with a GigaSMART engine port, as follows:

(config) # gsgroup alias grp2 port-list 1/8/e2

To display the gsgroup configuration, use the following CLI command:

(config) # show gsgroup

The **e** port references the GigaSMART line card or module.

Ex 1, Step 5: Configure the gsop

Define a gsop to enable NetFlow Generation, as follows:

(config) # gsop alias gsop2 flow-ops netflow port-list grp2

To display the gsop configuration, use the following CLI command:

(config) # show gsop

Ex 1, Step 6: Configure the IP Interface with a Tool Port

Identify the collector port and associate the port with the IP interface alias. Configure the port as a tool port, where the NetFlow collector will be connected, as follows:

(config) # port 1/1/g3 type tool

(config) # port 1/1/g3 params admin enable

To display the port configuration, use the following CLI command:

(config) # show port

Ex 1, Step 7: Configure Exporter Associated to IP Interface Tool Port

Configure an IP interface with a tool port and associate the NetFlow Generation Exporter to the IP interface tool port, as follows:

Step	Description	Command
1.	Configure the IP interface. The IP address is for the NetFlow interface. The port list was defined in Ex 1, Step 5: Configure the gsop. Associate the exporter to the IP interface. You can associate multiple exporters to the IP interface. This was defined in Ex 1, Step 1: Configure the Exporter.	**(config) # ip interface alias test** **(config ip interface alias test) # attach 1/1/g3** **(config ip interface alias test) # ip address 1.1.1.1 /29** **(config ip interface alias test) # gw 1.1.1.2** **(config ip interface alias test) # mtu 9400** **(config ip interface alias test) # gsgroup add grp2** **(config ip interface alias test) # netflow-exporter add exp4** **(config ip interface alias test) # exit**
2.	Display the IP interface configuration.	**(config) # show ip interface**

Ex 1, Step 8: Configure GigaSMART Params to Add a Monitor

Update the GigaSMART parameters to include the NetFlow Monitor, as follows:

(config) # gsparams gsgroup grp2 netflow-monitor add mon2

The monitor (mon2) was defined in Ex 1, Step 3: Configure the Monitor. The GigaSMART group was defined in Ex 1, Step 4: Configure the gsgroup.

NOTE: Only one NetFlow Generation Monitor can be configured per gsgroup.

To display the GigaSMART parameters configuration, use the following CLI command:

(config) # show gsparams

Ex 1, Step 9: Configure Mapping Rules to Filter Packets

To add Flow Mapping rules to filter packets that are needed to run NetFlow, configure a map and associate the map to the IP interface with tool port, as follows:

Step	Description	Command
1.	Configure the map. (This is a first level map.)	**(config) # map alias map3** **(config map alias map3) # type regular byRule** **(config map alias map3) # use gsop gsop2** **(config map alias map3) # rule add pass ipver 4** **(config map alias map3) # from 1/1/x11** **(config map alias map3) # to 1/1/g3** **(config map alias map3) # exit** **(config) #**
2.	Display the map configuration.	**(config) # show map**

Example 2: NetFlow Generation Configuration

Starting in software version 4.2, NetFlow exporters can filter NetFlow records. The filtered NetFlow records are sent to the collectors.

In Example 2, there are three exporters, with filtering configured on two of them. Since the second exporter does not have any filtering configured, all the records are sent to the collector. In this example, there are also two tunnels and two maps. Both maps are first level maps.

Ex 2, Step 1: Configure the Exporter

Configure one or more NetFlow Generation Exporter(s), as follows:

Step	Description	Command
1.	Configure the first exporter.	**(config) # apps netflow exporter alias exp1** **(config apps netflow exporter alias exp1) # format netflow version ipfix** **(config apps netflow exporter alias exp1) # destination ip4addr 1.1.1.1** **(config apps netflow exporter alias exp1) # filter add pass ipv4 dst any value 1.1.1.1 255.255.255.248** **(config apps netflow exporter alias exp1) # filter add pass vlan id any value 1** **(config apps netflow exporter alias exp1) # filter add pass l4port dst any value 1** **(config apps netflow exporter alias exp1) # exit** **(config) #**
2.	Configure the second	**(config) # apps netflow exporter alias exp2**

Step	Description	Command
	exporter.	**(config apps netflow exporter alias exp2) # format netflow version ipfix** **(config apps netflow exporter alias exp2) # destination ip4addr 2.2.2.2** **(config apps netflow exporter alias exp2) # transport udp 2055** **(config apps netflow exporter alias exp2) # dscp 10** **(config apps netflow exporter alias exp2) # exit** **(config) #**
3.	Configure the third exporter.	**(config) # apps netflow exporter alias exp3** **(config apps netflow exporter alias exp3) # format netflow version ipfix** **(config apps netflow exporter alias exp3) # destination ip4addr 3.3.3.3** **(config apps netflow exporter alias exp3) # filter add pass ipv4 dst any value 3.3.3.3 255.255.255.248** **(config apps netflow exporter alias exp3) # filter add pass vlan id any value 3** **(config apps netflow exporter alias exp3) # filter add pass l4port dst any value 3** **(config apps netflow exporter alias exp3) # exit** **(config) #**
4.	Display the exporter configuration.	**(config) # show apps netflow exporter**

Ex 2, Step 2: Configure the Record

Configure a NetFlow Generation Record, as follows:

Step	Description	Command
1.	Configure the record.	**(config) # apps netflow record alias rec1** **(config apps netflow record alias rec1) # netflow-version ipfix** **(config apps netflow record alias rec1) # match add ipv4 ttl** **(config apps netflow record alias rec1) # match add ipv6 traffic-class** **(config apps netflow record alias rec1) # collect add transport udp source-port(** **config apps netflow record alias rec1) # collect add transport tcp source-port**

Step	Description	Command
		(config apps netflow record alias rec1) # exporter add exp1 **(config apps netflow record alias rec1) # exporter add exp2** **(config apps netflow record alias rec1) # exporter add exp3** **(config apps netflow record alias rec1) # sampling set 1 in 30** **(config apps netflow record alias rec1) # exit** **(config) #**
2.	Display the record configuration.	**(config) # show apps netflow record**

To display the record configuration, use the following CLI command:

(config) # show apps netflow record alias rec1

Ex 2, Step 3: Configure the Monitor

Configure a NetFlow Generation Monitor and associate the NetFlow Generation Record to the specified NetFlow Generation Monitor, as follows:

Step	Description	Command
1.	Configure the monitor. **NOTE:** In this example, NetFlow sampling is set to multi-rate.	**(config) # apps netflow monitor alias mon1** **(config apps netflow monitor alias mon1) # record add rec1** **(config apps netflow monitor alias mon1) # sampling set multi-rate** **(config apps netflow monitor alias mon1) # exit** **(config) #**
2.	Display the monitor configuration.	**(config) # show apps netflow monitor**

Ex 2, Step 4: Configure the gsgroup

Configure a GigaSMART group and associate it with a GigaSMART engine port, as follows:

(config) # gsgroup alias grp port-list 1/8/e1

To display the gsgroup configuration, use the following CLI command:

(config) # show gsgroup

Ex 2, Step 5: Configure the gsop

Define a gsop to enable NetFlow Generation, as follows:

(config) # gsop alias gsop1 flow-ops netflow port-list grp

To display the gsop configuration, use the following CLI command:

(config) # show gsop

Ex 2, Step 6: Configure the IP Interface with a Tool Port

Identify the collector port and associate the port with the IP interface alias. Configure the port as a tool port, where the NetFlow collector will be connected, as follows:

(config) # port 1/1/g1 type tool
(config) # port 1/1/g1 params admin enable
(config) # port 1/1/g2 type tool
(config) # port 1/1/g2 params admin enable

To display the port configuration, use the following CLI command:

(config) # show port

Ex 2, Step 7: Configure Exporter Associated to IP Interface with Tool Port

Configure an IP interface with a tool port and associate the NetFlow Generation Exporter to the IP interface tool port, as follows:

Step	Description	Command
1.	Configure the first IP interface and associate two NetFlow exporters to the IP interface.	**(config) # ip interface alias test1** **(config ip interface alias test1) # attach 1/1/g1** **(config ip interface alias test1) # ip address 1.1.1.1 /29** **(config ip interface alias test1) # gw 1.1.1.2** **(config ip interface alias test1) # mtu 9400** **(config ip interface alias test1) # gsgroup add grp** **(config ip interface alias test1) # netflow-exporter add exp1,exp2** **(config ip interface alias test1) # exit**
2.	Configure the second IP interface and associate third NetFlow exporter to the IP interface.	**(config) # ip interface alias test2** **(config ip interface alias test2) # attach 1/1/g2** **(config ip interface alias test2) # ip address 4.4.4.3 /29** **(config ip interface alias test2) # gw 1.1.1.2** **(config ip interface alias test2) # mtu 9400** **(config ip interface alias test2) # gsgroup add grp** **(config ip interface alias test2) # netflow-exporter add exp3**

Step	Description	Command
		(config ip interface alias test2) # exit
3.	Display the IP interface configuration.	**(config) # show ip interface**

To display the IP interface configuration, use the following CLI command:

(config) # show ip interface

Ex 2, Step 8: Configure GigaSMART Parameters to Add a Monitor

Update the GigaSMART parameters to include the NetFlow Monitor, as follows:

(config) # gsparams gsgroup grp netflow-monitor add mon1

NOTE: Only one NetFlow Generation Monitor can be configured per gsgroup.

To display the GigaSMART parameters configuration, use the following CLI command:

(config) # show gsparams

...

Ex 2, Step 9: Configure Mapping Rules to Filter Packets

To add Flow Mapping rules to filter packets that are needed to run NetFlow, configure maps and associate the maps to the IP interface with tool ports, as follows:

Step	Description	Command
1.	Configure the first map. (This is a first level map.)	**(config) # map alias map1** **(config map alias map1) # type regular byRule** **(config map alias map1) # use gsop gsop1** **(config map alias map1) # rule add pass ipver 4** **(config map alias map1) # from 1/1/x1..x2** **(config map alias map1) # to 1/1/g1** **(config map alias map1) # exit** **(config) #**
2.	Configure the second map. (This is also a first level map.)	**(config) # map alias map2** **(config map alias map2) # type regular byRule** **(config map alias map2) # use gsop gsop1** **(config map alias map2) # rule add pass ipver 4** **(config map alias map2) # from 1/1/x3..x4** **(config map alias map2) # to 1/1/g2** **(config map alias map2) # exit**

Step	Description	Command
		(config) #
3.	Display the map configuration.	**(config) # show map**

Example 3: NetFlow Generation Configuration

Starting in software version 4.3.01, NetFlow supports both first level and second level maps. In Example 3, there are two maps. However, unlike Example 2, which has two first level maps, in this example, one map is a first level map and the other is a second level map. A virtual port is configured that directs traffic to the second level map.

The configuration of the GigaSMART operation in Example 3 differs from Example 1 and Example 2. The gsop sends traffic to APF first, and then to NetFlow.

In the first level map, the traffic matching the rule is sent to the virtual port. The same traffic is also sent to two tool ports (2/1/g2 and 2/1/g3).

In the second level map, the traffic from the virtual port matching the gsrule is sent to NetFlow and then to the IP interface with tool port, 2/1/g7.

Ex 3, Step 1: Configure the Exporter

Configure one or more NetFlow Generation Exporter(s), as follows:

Step	Description	Command
1.	Configure the exporter.	**(config) # apps netflow exporter alias exp1** **(config apps netflow exporter alias exp1) # format cef version 23** **(config apps netflow exporter alias exp1) # destination ip4addr 10.50.22.25** **(config apps netflow exporter alias exp1) # exit** **(config) #**

Ex 3, Step 2: Configure the Record

Configure a NetFlow Generation Record, as follows:

Step	Description	Command
1.	Configure the record.	**(config) # apps netflow record alias rec1** **(config apps netflow record alias rec1) #**

Step	Description	Command
		netflow-version ipfix **(config apps netflow record alias rec1) # match add ipv4 source address** **(config apps netflow record alias rec1) # match add ipv4 tos** **(config apps netflow record alias rec1) # collect add ipv4 protocol** **(config apps netflow record alias rec1) # collect add ipv4 source address** **(config apps netflow record alias rec1) # collect add interface input physical** **(config apps netflow record alias rec1) # exit** **(config) #**

Ex 3, Step 3: Configure the Monitor

Configure a NetFlow Generation Monitor and associate the NetFlow Generation Record to the specified NetFlow Generation Monitor, as follows:

Step	Description	Command
1.	Configure the monitor.	**(config) # apps netflow monitor alias mon1** **(config apps netflow monitor alias mon1) # record add rec1** **(config apps netflow monitor alias mon1) # exit** **(config) #**

Ex 3, Step 4: Configure the gsgroup

Configure a GigaSMART group and associate it with a GigaSMART engine port, as follows:

(config) # gsgroup alias grp port-list 2/1/e1

Ex 3, Step 5: Configure the Virtual Port

Configure a virtual port and associate it with the GigaSMART group, as follows:

(config) # vport alias vp1 gsgroup grp

Ex 3, Step 6: Configure the gsop

Define a gsop to enable NetFlow Generation, as follows:

(config) # gsop alias gsop_apf_netflow apf set flow-ops netflow port-list grp

Ex 3, Step 7: Configure the Tool Port

Identify the collector port and associate the port with the IP interface. Configure the port as a tool port, where the NetFlow collector will be connected, as follows:

```
(config) # port 2/1/g2..g3 type tool
(config) # port 2/1/g2..g3 params admin enable
(config) # port 2/1/g7 type tool
(config) # port 2/1/g7 params admin enable
```

Ex 3, Step 8: Configure Exporter Associated to IP Interface Tool Port

Configure an IP interface with a tool port and associate the NetFlow Generation Exporter to the IP interface tool port, as follows:

```
(config) # ip interface alias test
(config ip interface alias test) # attach 2/1/g7
(config ip interface alias test) # ip address 10.115.9.5 /21
(config ip interface alias test) # gw 10.115.8.1
(config ip interface alias test) # mtu 9400
(config ip interface alias test) # gsgroup add grp
(config ip interface alias test) # netflow-exporter add exp1
(config ip interface alias test) # exit
```

Ex 3, Step 9: Configure GigaSMART Parameters to Add a Monitor

Update the GigaSMART parameters to include the NetFlow Monitor, as follows:

```
(config) # gsparams gsgroup grp netflow-monitor add mon1
```

Ex 3, Step 10: Configure Mapping Rules to Filter Packets

To add Flow Mapping rules to filter packets that are needed to run NetFlow, configure maps and associate the maps to the IP interface with tool port, as follows:

Step	Description	Command
1.	Configure the first map. (This is a first level map.)	**(config) # map alias map1** **(config map alias map1) # type firstLevel byRule** **(config map alias map1) # rule add pass macdst 00:00:00:00:00:00 00:00:00:00:00:00** **(config map alias map1) # from 2/1/g1** **(config map alias map1) # to vp1,2/1/g2,2/1/g3** **(config map alias map1) # exit** **(config) #**
2.	Configure the second map. (This is a second level map.)	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule**

Step	Description	Command
		(config map alias map2) # use gsop gsop_apf_netflow **(config map alias map2) # gsrule add pass mac dst pos 1 value 00:00:00:00:00:00 00:00:00:00:00:00 (config map alias map2) # from vp1** **(config map alias map2) # to 2/1/g7** **(config map alias map2) # exit** **(config) #**

Example 4: NetFlow Generation Configuration

Starting in software version 4.3.01, NetFlow supports both first level and second level maps. In Example 4, there are three maps. One map is a first level map and the other two are second level maps. Two virtual ports are configured that direct traffic to the second level maps.

Two GigaSMART operations are configured. One gsop sends traffic to masking. The other gsop sends traffic to APF and then to NetFlow.

In the first level map, the traffic matching the rule is sent to two virtual ports. The same traffic is also sent to a tool port (11/1/g3).

In the first second level map, the traffic from the first virtual port, vp1, that matches the gsrule, is sent to masking and then to the tool port 11/1/g2.

In the next second level map, the traffic from the second virtual port, vp2, that matches the gsrule, is sent to NetFlow and then to the IP interface with tool port, 11/1/g4.

Ex 4, Step 1: Configure the Exporter

Configure one or more NetFlow Generation Exporter(s), as follows:

Step	Description	Command
1.	Configure the first exporter.	**(config) # apps netflow exporter alias exp1** **(config apps netflow exporter alias exp1) # format netflow version ipfix** **(config apps netflow exporter alias exp1) # destination ip4addr 10.50.22.25** **(config apps netflow exporter alias exp1) # exit** **(config) #**
2.	Configure the second exporter.	**(config) # apps netflow exporter alias exp2** **(config apps netflow exporter alias exp2) # format**

Step	Description	Command
		netflow version ipfix **(config apps netflow exporter alias exp2) # destination ip4addr 10.40.21.12** **(config apps netflow exporter alias exp2) # exit** **(config) #**

Ex 4, Step 2: Configure the Record

Configure a NetFlow Generation Record, as follows:

```
(config) # apps netflow record alias rec1
(config apps netflow record alias rec1) # netflow-version ipfix
(config apps netflow record alias rec1) # match add ipv4 source address
(config apps netflow record alias rec1) # match add ipv4 tos
(config apps netflow record alias rec1) # collect add ipv4 protocol
(config apps netflow record alias rec1) # collect add ipv4 source address
(config apps netflow record alias rec1) # collect add interface input physical
(config apps netflow record alias rec1) # exit
(config) #
```

Ex 4, Step 3: Configure the Monitor

Configure a NetFlow Generation Monitor and associate the NetFlow Generation Record to the specified NetFlow Generation Monitor, as follows:

```
(config) # apps netflow monitor alias mon1
(config apps netflow monitor alias mon1) # record add rec1
(config apps netflow monitor alias mon1) # cache timeout active 2
(config apps netflow monitor alias mon1) # cache timeout inactive 2
(config apps netflow monitor alias mon1) # exit
(config) #
```

Ex 4, Step 4: Configure the gsgroup

Configure GigaSMART groups and associate them with a GigaSMART engine port, as follows:

```
(config) # gsgroup alias grp1 port-list 11/3/e1
(config) # gsgroup alias grp2 port-list 11/3/e2
```

Ex 4, Step 5: Configure the Virtual Port

Configure virtual ports and associate them with the GigaSMART group, as follows:

```
(config) # vport alias vp1 gsgroup grp1
(config) # vport alias vp2 gsgroup grp2
```

Ex 4, Step 6: Configure the gsop

Define the GigaSMART operation to enable masking, as follows:

```
(config) # gsop alias gsop_mask_aa apf set masking protocol none offset 50 pattern aa length 100 port-list grp1
```

Define the GigaSMART operation to enable NetFlow, as follows:

```
(config) # gsop alias gsop_apf_netflow apf set flow-ops netflow port-list grp2
```

Ex 4, Step 7: Configure a Tool Port

Identify the collector port and associate the port with the IP interface alias. Configure the port as a tool port, where the NetFlow collector will be connected, as follows:

```
(config) # port 11/1/g2..g4 type tool
(config) # port 11/1/g2..g4 params admin enable
```

Ex 4, Step 8: Configure Exporter Associated to IP Interface Tool Port

Configure an IP interface with a tool port and associate the NetFlow Generation Exporter to the IP interface tool port, as follows:

```
(config) # ip interface alias test
(config ip interface alias test) # attach 11/1/g4
(config ip interface alias test) # ip address 10.115.9.6 /21
(config ip interface alias test) # gw 10.115.8.1
(config ip interface alias test) # mtu 9400
(config ip interface alias test) # gsgroup add grp2
(config ip interface alias test) # netflow-exporter add exp2
(config ip interface alias test) # exit
```

Ex 4, Step 9: Configure GigaSMART Parameters to Add a Monitor

Update the GigaSMART parameters to include the NetFlow Monitor, as follows:

```
(config) # gsparams gsgroup grp2 netflow-monitor add mon1
```

Ex 4, Step 10: Configure Mapping Rules to Filter Packets

To add Flow Mapping rules to filter packets that are needed to run NetFlow, configure maps and associate the maps to the IP interface tool ports, as follows:

Step	Description	Command
1.	Configure the first map. (This is a first level map.)	**(config) # map alias map1** **(config map alias map1) # type firstLevel byRule** **(config map alias map1) # rule add pass macdst 00:00:00:00:00:00 00:00:00:00:00:00** **(config map alias map1) # from 11/1/g1**

Step	Description	Command
		(config map alias map1) # to vp1,vp2,11/1/g3 **(config map alias map1) # exit** **(config) #**
2.	Configure the second map. (This is a second level map.)	**(config) # map alias map2** **(config map alias map2) # type secondLevel byRule** **(config map alias map2) # use gsop gsop_mask_aa** **(config map alias map2) # gsrule add pass mac dst pos 1 value 00:00:00:00:00:00 00:00:00:00:00:00** **(config map alias map2) # from vp1** **(config map alias map2) # to 11/1/g2** **(config map alias map2) # exit** **(config) #**
3.	Configure the third map. (This is also a second level map.)	**(config) # map alias map3** **(config map alias map3) # type secondLevel byRule** **(config map alias map3) # use gsop gsop_apf_netflow** **(config map alias map3) # gsrule add pass mac dst pos 1 value 00:00:00:00:00:00 00:00:00:00:00:00** **(config map alias map3) # from vp2** **(config map alias map3) # to 11/1/g4** **(config map alias map3) # exit** **(config) #**
4.	Display the following statistics: ▪ Exporter statistics ▪ Monitor statistics ▪ IP interface statistics	**(config) # show apps netflow exporter stats** **(config) # show apps netflow monitor stats** **(config) # show ip interface stats**

NetFlow Generation Configuration Modification and Removal

There may be instances where a NetFlow Generation configuration may require alteration by modifying a NetFlow Generation Monitor Configuration or a NetFlow Generation Record Configuration. It may further require that the configuration be removed entirely. In such instances, refer to the following.

Modify a NetFlow Generation Monitor Configuration

This example shows the modification of a NetFlow Generation Monitor configuration.

1. Unlink the monitor from gsparams.

   gsparams gsgroup <gsgroup> netflow-monitor delete

2. Modify the monitor parameters.

   apps netflow monitor alias <monitor> record delete <record> <change monitor parameters>

3. Re-add the record to the monitor.

   apps netflow monitor alias <monitor> record add <record>

4. Re-add the monitor to gsparams for changes to take affect.

   gsparams gsgroup <gsgroup> netflow-monitor add <monitor>

Modify a NetFlow Generation Record Configuration

This example shows the modification of a NetFlow Generation Record configuration.

1. Unlink the monitor from gsparams.

   gsparams gsgroup <gsgroup> netflow-monitor delete

2. Modify the record bound to the monitor.

   apps netflow record alias <record> <change record parameters>

3. Re-add the monitor to gsparams for changes in record to take affect.

   gsparams gsgroup <gsgroup> netflow-monitor add <monitor>

Remove a NetFlow Generation Configuration

Use the following commands to remove a NetFlow Generation Configuration:

gsparams gsgroup <gsgroup> netflow-monitor delete no map alias <map> no tunneled-port port <port> no apps netflow monitor alias <monitor> no apps netflow record alias <record> no apps netflow exporter alias <exporter>

V5 Fixed Record Template

NetFlow v5 records have a template of fixed fields that cannot be edited. The template contains Match/Key and Collect/Non-Key elements. It has an alias of **predefined_netflow_v5_record**.

To display the template, use the following CLI command:

(config) # show apps netflow record alias predefined_netflow_v5_record

GigaSMART Load Balancing

Load balancing distributes GigaSMART outgoing traffic to multiple tool ports or multiple tunnel endpoint destinations. In this way, traffic processed by GigaSMART is shared.

- Stateful load balancing distributes GigaSMART processed traffic to multiple tool ports or tunnel endpoints based on GigaSMART application-specific flow sessions.

- Stateless load balancing distributes GigaSMART processed traffic to multiple tool ports or tunnel endpoints based on hash values generated from predefined protocol fields in the packet.
- Enhanced load balancing

Load balancing operations to tool ports can be assigned to GigaSMART groups consisting of multiple engine ports. Refer to *Groups of GigaSMART Engine Ports* on page 406 for details.

Refer to the Load Balancing section in the GigaVUE Fabric Management Guide.

Stateful Loadbalancing

Refer to the following examples:

- Example 1: GigaSMART Stateful Load Balancing
- Example 2: GigaSMART Stateful Load Balancing

For an example of GTP load balancing in a cluster, refer to Example 6: GigaSMART GTP Load Balancing in a Cluster.

For an example of load balancing on L2GRE encapsulation tunnel, refer to Example 2 – GigaSMART L2GRE Tunnel Encap Stateful LB.

Example 1: GigaSMART Stateful Load Balancing

Example 1 configures stateful load balancing of GigaSMART GTP traffic among tool ports 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4 based on bandwidth with different weights for each port. The same subscriber (imsi) traffic will be forwarded to the same tool port. GTP-c packets are replicated to all tool ports.

Weight determines the traffic sent to a particular port. The weight of the individual ports must be less than 100. The combined value of the ports can be greater than 100, as the actual load balancing ratio is computed with individual values divided by the combined value. For example, if there are three ports with the following port weights:

- 1/1/x6 50
- 1/1/x7 60
- 1/1/x8 90

Then distribution is as follows:

- For the port 1/1/x6, the distribution will be 50/(200)*100 = 25%
- For the port 1/1/x7 60, the distribution will be 60/200*100 = 30%
- For the port 1/1/x8 90, the distribution will be 90/200 *100 = 45%

The following table provides step-by-step configuration for Stateful Load Balancing.

Step	Description	Command
1.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
2.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
3.	Specify weights for each tool port.	**(config) # port-group alias portgrp1 weight 1/1/x6 5** **(config) # port-group alias portgrp1 weight 1/1/x7 10** **(config) # port-group alias portgrp1 weight 1/2/x3 20** **(config) # port-group alias portgrp1 weight 1/2/x4 10**
4.	Create a GigaSMART group and specify a port.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1**
5.	Enable replicate GTP-c packets to all tool ports in the load balancing port group.	**(config) # gsparams gsgroup gsgrp1 lb replicate-gtp-c enable**
6.	Create a GSOP, including GTP application and load balancing metric.	**(config) # gsop alias gsop1 flow-ops flow-filtering gtp lb app gtp metric wt-lt-bw port-list gsgrp1**
7.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
8.	Create an ingress (first level) map. Note the following: ▪ You can specify only one port group as part of the map tool port in the **to** statement. ▪ You can define the same load balancing port group in multiple maps, however, the load balancing metrics defined in the GSOPs on those maps have to be the same. ▪ You cannot use a shared collector map for load	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass portdst 2123** **(config map alias map11) # rule add pass portdst 2152** **(config map alias map11) # exit** **(config) #**

Step	Description	Command
	balancing. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	
9.	Create a second level map.	**(config) # map alias map22** **(config map alias map22) # type secondLevel flowFilter** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to portgrp1** **(config map alias map22) # flowrule add pass gtp imsi 234567*** **(config map alias map22) # exit** **(config) #**
10.	Display load balancing statistics.	**(config) # show load-balance port-group stats alias portgrp1**
11.	Clear load balancing statistics, including Total Bytes, Total Packets, and Total Sessions. **Note**: Active Sessions will not be cleared.	(config) # clear load-balance port-group stats all

Example 2: GigaSMART Stateful Load Balancing

Example 2 configures stateful load balancing of GigaSMART GTP traffic among tool ports 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4 based on hashing of the imei value. The same device ID (imei) traffic will be forwarded to the same tool port. GTP-c packets are replicated to all tool ports.

Step	Description	Command
1.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
2.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
3.	Create a GigaSMART group and specify ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1**
4.	Enable replicate GTP-c packets to all tool ports in the load balancing port group.	**(config) # gsparams gsgroup gsgrp1 lb replicate-gtp-c enable**
5.	Create a GSOP, including GTP	**(config) # gsop alias gsop1 flow-ops flow-**

Step	Description	Command
	application and load balancing metric.	**filtering gtp lb app gtp metric hashing key imei port-list gsgrp1**
6.	Create a virtual port and associate it with the GigaSMART group.	**(config) # vport alias vp1 gsgroup gsgrp1**
7.	Create an ingress (first level) map. Note the following: ▪ You can specify only one port group as part of the map tool port in the **to** statement. ▪ You can define the same load balancing port group in multiple maps, however, the load balancing metrics defined in the GSOPs on those maps have to be the same. ▪ You cannot use a shared collector map for load balancing. **Note**: In the rules, 2123 is GTP-c traffic and 2152 is GTP-u traffic.	**(config) # map alias map11** **(config map alias map11) # type firstLevel byRule** **(config map alias map11) # from 1/1/x1** **(config map alias map11) # to vp1** **(config map alias map11) # rule add pass portdst 2123** **(config map alias map11) # rule add pass portdst 2152** **(config map alias map11) # exit** **(config) #**
8.	Create a second level map.	**(config) # map alias map22** **(config map alias map22) # type secondLevel flowFilter** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to portgrp1** **(config map alias map22) # flowrule add pass gtp imsi 234567*** **(config map alias map22) # exit** **(config) #**
9.	Display load balancing statistics.	**(config) # show load-balance port-group stats alias portgrp1**

Use the following command to display load balancing statistics:

(config) # show load-balance port-group stats alias portgrp1

Stateless Loadbalancing

Refer to the following examples:

- Example 1: GigaSMART Stateless Load Balancing
- Example 2: GigaSMART Stateless Load Balancing
- Example 3: GigaSMART Stateless Load Balancing

For an example of load balancing on L2GRE encapsulation tunnel, refer to Example 3 – GigaSMART L2GRE Tunnel Encap Stateless LB.

Example 1: GigaSMART Stateless Load Balancing

Example 1 configures stateless load balancing of traffic among tool ports 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4 after slicing the packet to an offset of 70 bytes.

Step	Description	Command
1.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
2.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
3.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
4.	Create a GSOP, with load balancing.	**(config) # gsop alias lbiponlyouter slicing protocol none offset 70 lb hash ip-only outer port-list gsgrp1**
5.	Create a first level map. Note the following: ▪ You can specify only one port group as part of the map tool port in the **to** statement. ▪ You can define the same load balancing port group in multiple maps, however, the load balancing metrics defined in the GSOPs on those maps have to be the same. ▪ You cannot use a shared collector map for load	**(config) # map alias map1** **(config map alias map1) # type regular byRule** **(config map alias map1) # from 1/1/x1** **(config map alias map1) # use gsop lbiponlyouter** **(config map alias map1) # to portgrp1** **(config map alias map1) # rule add pass ipver4** **(config map alias map1) # exit (config) #**

Step	Description	Command
	balancing.	
6.	Display load balancing statistics.	**(config) # show load-balance port-group stats alias portgrp1**
7.	Clear load balancing statistics, including Total Bytes and Total Packets.	(config) # clear load-balance port-group stats all
8.	Display load balancing statistics. **Note**: Since stateless load balancing is on a packet-by-packet basis, it does not have sessions. So for stateless load balancing, Total Sessions and Active Sessions will always be zero (0).	**(config) # show load-balance port-group stats alias portgrp1**

Example 2: GigaSMART Stateless Load Balancing

Example 2 configures stateless load balancing of GTP traffic among tool ports 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4. Data packets with the same GTP-u tunnel ID will be forwarded to the same tool port.

Step	Description	Command
1.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
2.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
3.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
4.	Create a GSOP, including load balancing metric.	**(config) # gsop alias gsop1 lb hash gtpu-teid port-list gsgrp1**
5.	Create first level maps. Note the following: ▪ You can specify only one port group as part of the map tool port in the **to** statement. ▪ You can define the same load balancing port group in multiple maps,	**(config) # map alias map1** **(config map alias map1) # type regular byRule** **(config map alias map1) # from 1/1/x1** **(config map alias map1) # to portgrp1** **(config map alias map1) # rule add pass protocol udp portdst 2123** **(config map alias map1) # exit** **(config) # map alias map2** **(config map alias map2) # type regular byRule**

Step	Description	Command
	however, the load balancing metrics defined in the GSOPs on those maps have to be the same. ▪ You cannot use a shared collector map for load balancing.	**(config map alias map2) # from 1/1/x1** **(config map alias map2) # use gsop gsop1** **(config map alias map2) # to portgrp1** **(config map alias map2) # rule add pass protocol udp portdst 2152** **(config map alias map2) # exit** **(config) #**
6.	Display load balancing statistics. **Note**: Since stateless load balancing is on a packet-by-packet basis, it does not have sessions. So for stateless load balancing, Total Sessions and Active Sessions will always be zero (0).	**(config) # show load-balance port-group stats alias portgrp1**

Example 3: GigaSMART Stateless Load Balancing

Example 3 configures stateless load balancing of HTTP on GTP traffic among tool ports 1/1/x6, 1/1/x7, 1/2/x3, and 1/2/x4. Data packets with the same inner IP will be forwarded to the same tool port.

Step	Description	Command
1.	Create a port group and specify the tool ports for load balancing.	**(config) # port-group alias portgrp1 port-list 1/1/x6,1/1/x7,1/2/x3,1/2/x4**
2.	Enable load balancing on the port group.	**(config) # port-group alias portgrp1 smart-lb enable**
3.	Configure a GigaSMART group and associate it with GigaSMART engine ports.	**(config) # gsgroup alias gsgrp1 port-list 1/3/e1,1/3/e2**
4.	Create a GSOP, including load balancing metric.	**(config) # gsop alias gsop1 lb hash ip-only inner port-list gsgrp1**
5.	Create first level and second level maps. Note the following: ▪ You can specify only one port group as part of the map tool port in the **to** statement. ▪ You can define the same	**(config) # map alias map1** **(config map alias map1) # type regular byRule** **(config map alias map1) # from 1/1/x1** **(config map alias map1) # to portgrp1** **(config map alias map1) # rule add pass protocol udp portdst 2123** **(config map alias map1) # exit** **(config) # map alias map2**

Step	Description	Command
	load balancing port group in multiple maps, however, the load balancing metrics defined in the GSOPs on those maps have to be the same. ▪ You cannot use a shared collector map for load balancing.	**(config map alias map2) # type firstLevel byRule** **(config map alias map2) # from 1/1/x1** **(config map alias map2) # to vp1** **(config map alias map2) # rule add pass protocol udp portdst 2152** **(config map alias map2) # exit** **(config) #map alias map22** **(config map alias map22) # type secondLevel byRule** **(config map alias map22) # from vp1** **(config map alias map22) # use gsop gsop1** **(config map alias map22) # to portgrp1** **(config map alias map22) # gsrule add pass l4port dst pos 2 value 80** **(config map alias map22) # exit** **(config) #**
6.	Display load balancing statistics. **Note**: Since stateless load balancing is on a packet-by-packet basis, it does not have sessions. So for stateless load balancing, Total Sessions and Active Sessions will always be zero (0).	**(config) # show load-balance port-group stats alias portgrp1**

Enhanced Load Balancing

Example 1: Enhanced Load Balancing

In this example GTP-c traffic is sent to tool ports 1/1/x5, 1/1/x6. Enhanced load balancing is used to send GTP-u traffic to tool ports 1/1/x5, 1/1/x6, 2/1/x15 and 2/2/x16 based on inner IP.

NOTE: Optional configuration for MPLS traffic handling. Inner IP (version 4) is located 70 bytes from the beginning of the packet.

Task	Description	UI Steps
1.	Create a Port Group for GTP-c traffic and specify the tool ports.	# port-group alias pg1 port-list 1/1/x5,1/1/x6 exit
2.	Create a Port Group for GTP-u traffic	# port-group alias pg2

Task	Description	UI Steps
	and specify the tool ports for enhanced load balancing.	port-list 1/1/x5,1/1/x6,2/1/x15,2/2/x16 smart-lb enable exit
3.	Create a GigaSMART Group and associate the GigaSMART engine port (s).	# gsgroup alias gsgrp1 port-list 1/1/e1
4.	Create an enhanced load balance metric for GTPu traffic and distribute traffic based on inner IP.	# apps enhanced-lb alias elb-gtpu hash-field add ip inner exit
5.	Create a GigaSMART operation for GTP-c traffic.	# gsop alias gsop1 apf set port-list gsgrp1 exit
6.	Create a GigaSMART operation for GTP-u traffic and include the enhanced load balance metric.	# gsop alias gsop2 apf set lb elb elb-gtpu port-list gsgrp1 exit
7.	Create a virtual port and associate it with the GigaSMART group.	# vport alias vp1 gsgroup gsgrp1
8.	Create an ingress first level map to direct traffic from the network ports 1/1/x1 and 1/1/x2 to the virtual port based on IP version.	# map alias map11 from 1/1/x1,1/1/x2 to vp1 rule add pass ipver4 rule add pass ipver6 exit
9.	Create an egress second level map to process GTP-c traffic	# map alias map21 from vp1 use gsop gsop1
10.	Create an egress second level map to process GTP-u traffic	# map alias map22 from vp1 use gsop gsop2 to pg2 gsrule add pass l4port dst pos 1 value 2152 exit

Task	Description	UI Steps
11.	Display port statistics for GTP-c traffic	# show port stats port-list 1/1/x5..x6
12.	Display load balancing statistics for GTP-u traffic	# show load-balance port-group stats alias pg2

Example 2: Enhanced Load Balancing

In this example enhanced load balancing of Non GTP traffic and (subsequent) fragmented packets to are sent GTP tool ports 1/1/x5, 1/1/x6, 2/1/x15 and 2/2/x16 based on outer IP.

Task	Description	UI Steps
1.	Create a Port Group for non GTP and (subsequent) fragmented traffic	# port-group alias pg3 port-list 1/1/x5,1/1/x6,2/1/x15,2/2/x16 smart-lb enable exit
2.	Create a GigaSMART Group and associate the GigaSMART engine port (s)	# gsgroup alias gsgrp1 port-list 1/1/e1
3.	Create an enhanced load balance metric for non GTP IP traffic and distribute traffic based on outer IP	# apps enhanced-lb alias elb-ip hash-field add ip outer exit
4.	Create a GigaSMART operation for non-GTP and fragmented traffic and include the enhanced load balance metric	# gsop alias gsop3 apf set lb elb elb-ip port-list gsgrp1 exit
5.	Create a virtual port and associate it with the GigaSMART group	# vport alias vp1 gsgroup gsgrp1
6.	Create an ingress first level map to direct traffic from the network ports 1/1/x1 and 1/1/x2 to the virtual port based on IP version	# map alias map11 from 1/1/x1,1/1/x2 to vp1 rule add pass ipver4 rule add pass ipver6 exit
7.	Create an egress second level map to process non GTP and fragmented traffic	# map alias map23 from vp1 use gsop gsop3

Task	Description	UI Steps
		to pg3 gsrule add pass ipver pos 1 value 4 Enhanced Load Balancing 9 gsrule add pass ipver pos 1 value 6 exit
8.	Display load balancing statistics for Non GTP and fragmented traffic	# show load-balance port-group stats alias pg3

GigaSMART MPLS Traffic Performance Enhancement

The GigaSMART MPLS traffic performance enhancement provides a method to improve GigaSMART packet processing for MPLS traffic and other traffic having Layer 2 encapsulation, such as L2GRE or VNTag. This type of traffic has a header in the packet between the MAC address and the IP address. Figure 25MPLS Header Between MAC and IP Address in Packet shows the MPLS example.

Figure 25 *MPLS Header Between MAC and IP Address in Packet*

Refer to the following examples:

- Flow Masking Example 1
- Flow Masking Example 2

Flow Masking Example 1

In Example 1 packets are expected to have two MPLS labels before the IP header, and no VLAN tag between the MAC and MPLS headers. IP addresses will be used to identify the flows.

The offset will be the sum of the following: 14 bytes for the MAC address + 8 bytes for the MPLS headers +12 bytes offset from the beginning of the IP header = 34 bytes.

The length will be the sum of the following: 4 bytes for ipsrc + 4 bytes for ipdst = 8 bytes.

Use the following CLI command syntax to configure Example 1:

(config) # gsparams gsgroup gsgrp1 flow-mask enable offset 34 length 8

Flow Masking Example 2

In Example 2, packets are expected to have one VLAN tag and two MPLS labels before the IP header. IP addresses will be used to identify the flows.

The offset will be the sum of the following: 14 bytes for the MAC address + 4 bytes for the VLAN tag + 8 bytes for the MPLS headers +12 bytes offset from the beginning of the IP header = 38 bytes.

The length will be the sum of the following: 4 bytes for ipsrc + 4 bytes for ipdst = 8 bytes.

Use the following CLI command syntax to configure Example 2:

(config) # gsparams gsgroup gsgrp1 flow-mask enable offset 38 length 8

GigaSMART Internet Content Adaptation Protocol (ICAP)

GigaSMART ICAP Client app can integrate inline iSSL with the DLP ICAP server by deploying the ICAP Client app as an inline tool. ICAP app will be an inline tool in the GigaSMART engine. The decrypted traffic from the inline SSL will be sent to the ICAP client through the configured inline network. The decrypted traffic will then be sent to the ICAP server for inspection.

The following is a configuration example of the ICAP Client.

For details on the CLI commands used in the following examples, refer to the following commands in the reference section:

- inline-network
- ip interface
- gsgroup
- gsop
- port
- map

Step	Description	Command
1.	Configure inline network.	(**config) # port 1/1/x5..x6 type inline-network** **(config) # port 1/1/x3..x4 type inline-network** **(config) #** **(config) # inline-network alias in1** **(config inline-network alias in1) # pair net-a 1/1/x5 and net-b 1/1/x6**

Step	Description	Command
		(config inline-network alias in1) # exit **(config) #** **(config) # inline-network alias in2** **(config inline-network alias in2) # pair net-a 1/1/x3 and net-b 1/1/x4** **(config inline-network alias in2) # exit** **(config) #** **(config) # inline-network-group alias ing1** **(config inline-network-group alias ing1) # network-list in2,in1** **(config inline-network-group alias ing1) # exit** **(config) #** **(config) # inline-network alias in1 traffic-path to-inline-tool** **(config) # inline-network alias in2 traffic-path to-inline-tool** **(config) #**
2.	Configure the IP interface.	**(config) # ip interface alias ip1** **(config ip interface alias ip1) # attach 1/1/x9** **(config ip interface alias ip1) # gsgroup add gsg3e1** **(config ip interface alias ip1) # ip ad 1.1.1.1 /24** **(config ip interface alias ip1) # gw 1.1.1.2** **(config ip interface alias ip1) # mtu 1500** **(config ip interface alias ip1) # exit**
3.	Configure the ICAP server.	**(config) # apps icap server alias server1** **(config apps icap server alias server1) # l3 address 1.1.1.2** **(config apps icap server alias server1) # l4 port 1344** **(config apps icap server alias server1) # service-url reqmod icap://1.1.1.2/reqmod** **(config apps icap server alias server1) # service-url respmod icap://1.1.1.2/respmod** **(config apps icap server alias server1) # service-url options icap://1.1.1.2/options** **(config apps icap server alias server1) # exit**
4.	Configure the ICAP server group.	**(config) # apps icap server-group alias server_group** **(config apps icap server-group alias server_group) # server-list server1** **(config apps icap server-group alias server_group) # exit** **(config) #**

Step	Description	Command
5.	Configure a GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias gsg3e1 port-list 1/3/e1**
6.	Configure the ICAP profile.	**(config) # apps icap profile alias icap1** **(config apps icap profile alias icap1) # http-request-buffer 10000 exceed bypass** **(config apps icap profile alias icap1) # preview 4** **(config apps icap profile alias icap1) # reqmod enable** **(config apps icap profile alias icap1) # respmod disable** **(config apps icap profile alias icap1) # server-group server_group** **(config apps icap profile alias icap1) # src-l4port 10000 to 60000** **(config apps icap profile alias icap1) # exit** **(config) #**
7.	Configure Gsop.	**(config) # gsop alias icap_1 icap icap1 port-list gsg3e1**
8.	Configure the map.	**(config) # map alias map1** **(config map alias map1) # from ing1** **(config map alias map1) # to 1/1/x9** **(config map alias map1) # use gsop icap_1** **(config map alias map1) # rule add pass macdst 00:00:00:00:00:00 00:00:00:00:00:00 bidir** **(config map alias map1) # type regular inlinePair** **(config map alias map1) # exit** **(config) #**

GigaSMART SSL Decryption for Out-of-Band Tools

GigaVUE HC Series nodes support Secure Sockets Layer (SSL) decryption. SSL is a cryptographic protocol that adds security to TCP/IP communications such as Web browsing and email. The protocol allows the transmission of secure data between a server and client who both have the keys to decode the transmission and the certificates to verify trust between them. Passive SSL decryption delivers decrypted traffic to out-of-band tools that can then detect threats entering the network.

SSL decryption is a pillar of the GigaSECURE Security Delivery Platform. For an overview of GigaSECURE, refer to the "*GigaSECURE Security Delivery Platform"* section in the *GigaVUE Fabric Management Guide*.

Configure Passive SSL Decryption Examples

The following sections provide examples of Passive SSL decryption. Refer to the following:

- Example 1: Passive SSL Decryption with a Regular Map and SSL private key

- Example 2: Passive SSL Decryption with a Regular Map and SSL server certificate
- Example 3: Passive SSL Decryption with De-duplication
- Other Usage Examples

For details on the CLI commands used in the following sections, refer to apps ssl, gsparams, and gsop in the reference section.

Example 1: Passive SSL Decryption with a Regular Map and SSL private key

In Example 1, a regular map is configured to use with Passive SSL decryption.

Step	Description	Command
1.	Upload a key and create a service. Refer to *Working with Keys and Services* on page 732.	**(config) # apps ssl key alias key1 download type private-key url https://keyserver.domain.com/path/keyfile.pem** **(config) # apps ssl service alias service1 server-ip 192.168.1.1 server-port 443**
2.	Configure a GigaSMART group.	**(config) # gsgroup alias gsgrp1 port-list 1/1/e1**
3.	Specify the GigaSMART group alias.	**(config) # gsparams gsgroup gsgrp1**
4.	Specify a failover action.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt decrypt-fail-action drop**
5.	Configure session timeouts, in seconds.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt pending-session-timeout 60** **(config gsparams gsgroup gsgrp1) # ssl-decrypt session-timeout 300** **(config gsparams gsgroup gsgrp1) # ssl-decrypt tcp-syn-timeout 20**
6.	Configure cache timeouts, in seconds.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt key-cache-timeout 9000** **(config gsparams gsgroup gsgrp1) # ssl-decrypt ticket-cache-timeout 9000**
7.	Configure a key/service mapping that maps how a key is assigned to an IP address of a server.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt key-map add service service1 key key1**
8.	Enable Passive SSL decryption.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt enable**
9.	Exit the GigaSMART group configuration mode.	**(config gsparams gsgroup gsgrp1) # exit** **(config) #**
10.	Configure a GigaSMART	**(config) # gsop alias gdssl1 ssl-decrypt in-port any**

Step	Description	Command
	operation for Passive SSL decryption.	**out-port auto port-list gsgrp1**

In the previous step, **gdssl1** is the alias for a GigaSMART operation, **in-port** specifies the destination port on which to listen, **out-port** specifies the destination port on which to send decrypted traffic, and **port-list** is set to the GigaSMART group alias previously configured. The **in-port** and **out-port** arguments can also be a port number between 1 and 65535.

Next, configure a traffic map, as follows:

Step	Description	Command
1.	Specify a map alias (m1) and specify the map type and subtype.	**(config) # map alias m1** **(config map alias m1) # type regular byRule**
2.	Specify the GigaSMART operation alias (gdssl1) as part of the map. This applies the associated GigaSMART functionality to packets matching a rule in the map.	**(config map alias m1) # use gsop gdssl1**
3.	Specify a map rule.	**(config map alias m1) # rule add pass ipver 4**
4.	Specify the destination for packets matching this map.	**(config map alias m1) # to 1/1/g2**
5.	Specify the source port(s) for this map.	**(config map alias m1) # from 1/1/g1**
6.	Exit the map prefix mode.	**(config map alias m1) # exit** **(config) #**
7.	Display the configuration.	**(config) # show gsop** **(config) # show map** **(config) # show gsparams**

Example 2: Passive SSL Decryption with a Regular Map and SSL server certificate

In Example 2, a regular map is configured to use with Passive SSL decryption.

Step	Description	Command
1.	Upload a key and certificate in apps keystore. For more	**(config) # apps keystore rsa key_name certificate key-str ***

Step	Description	Command
	information, refer toapps keystore.	
2.	Configure a GigaSMART group.	**(config) # gsgroup alias gsgrp1 port-list 1/1/e1**
3.	Specify the GigaSMART group alias.	**(config) # gsparams gsgroup gsgrp1**
4.	Specify a failover action.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt decrypt-fail-action drop**
5.	Configure session timeouts, in seconds.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt pending-session-timeout 60** **(config gsparams gsgroup gsgrp1) # ssl-decrypt session-timeout 300** **(config gsparams gsgroup gsgrp1) # ssl-decrypt tcp-syn-timeout 20**
6.	Configure cache timeouts, in seconds.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt key-cache-timeout 9000** **(config gsparams gsgroup gsgrp1) # ssl-decrypt ticket-cache-timeout 9000**
7.	Enable Passive SSL decryption.	**(config gsparams gsgroup gsgrp1) # ssl-decrypt enable**
8.	Exit the GigaSMART group configuration mode.	**(config gsparams gsgroup gsgrp1) # exit** **(config) #**
9.	Configure a GigaSMART operation for Passive SSL decryption.	**(config) # gsop alias gdssl1 ssl-decrypt in-port any out-port auto port-list gsgrp1**

You can use the server certificate, fetched from the initial client-server negotiation and use the same to uniquely identify the corresponding private key configured in Gigamon box.

The server-ip to private key service mapping is still supported. When none of the existing server-ip matches with the session IP, then the server certificate will be looked upon for certificate match

Example 3: Passive SSL Decryption with De-duplication

In Example 2, the configuration steps are the same except when you configure a GigaSMART operation you send the decrypted traffic to de-duplication for additional filtering, as follows:

(config) # gsop alias gdssl1 ssl-decrypt in-port any out-port auto dedup set port-list gsgrp1

Other Usage Examples

Two typical usage examples are as follows:

- Use map rules to filter on the IP address of the server and send everything to GigaSMART. Configure a GigaSMART operation to listen on the **in-port** used by the server. The GigaSMART will drop other traffic.
- Use map rules to filter on the IP address of the server and **in-port** and send specific port traffic to the GigaSMART. Configure a GigaSMART operation to listen on **in-port any**.

Thales Luna HSM for SSL Decryption for iSSL

> **NOTE:** Starting in release 6.14.00, Entrust/nCipher HSM integration is deprecated. Gen2 DSSL HSMs (nCipher v1.0) and Gen3 HSMs using nCipher 12.6 (`nshield_12_60`) are no longer supported, and upgrades are blocked if any nCipher HSM configuration is present.

Hardware Security Modules (HSMs) are specialized systems that logically and physically safeguard cryptographic operations and cryptographic keys. HSMs protect sensitive data from being stolen by providing a highly secure operation structure. HSMs are comprehensive, self-contained solutions for cryptographic processing, key generation, and key storage. The hardware and firmware (i.e., software) required for these functions are automatically included in these appliances.

Some enterprises where security is paramount use Thales Luna HSM to keep the sensitive information such as private keys safe.

The following is the configuration example of the Luna type HSM.

For details on the CLI commands used in the following examples, refer to the following commands in the reference section:

- apps hsm
- apps hsm-group
- apps keystore
- apps ssl
- gigasmart
- gsgroup
- gsop
- gsparams
- map

Configure Thales-Luna HSM:

Step	Description	Command
1.	Configure the stack port	**(config) # gigasmart engine 1/2/e1 interface eth2**

Step	Description	Command
	interface IP address	**10.117.72.60 255.255.248.0 gateway 10.115.71.1 dns 10.10.1.20 mtu 1500** or **config # gigasmart engine 1/2/e1 interface dhcp** **NOTE:** Make sure to configure the stack port first before you enable the HSM group in gsparams.
2.	Configure at least one HSM by specifying an alias, a static IP address, type of HSM, and port number. The server-password and partition-password should be encrypted using the keychain password in the Keystore. The partition-password for all hsm within the same HA group should be the same.	**(config) # apps hsm alias hsm1 hsm-ip 10.126.62.11 hsm-port 1792 type luna-hsm server-username admin server-password ****** partition-label partition1 partition-password *******
3.	Create an HSM group alias and add at least one HSM to it.	**(config) # apps hsm-group alias hsm-set device-type luna-hsm hsm-alias add hsm1**
4.	Configure the key label of the private key residing on Thales-Luna HSM.	**(config) # apps keystore rsa server_key_001 private-key key-label server1_key_label type luna-hsm** **(config) # apps keystore ecdsa server_key_001 private-key key-label server1_key_label type luna-hsm**
5.	Configure an inline SSL profile. The profile specifies policy configuration, such as certificate handling and actions to take for the profile.	**(config) # apps inline-ssl profile alias sslprofile** **(config apps inline-ssl profile alias sslprofile) # certificate expired drop** **(config apps inline-ssl profile alias sslprofile) # certificate invalid decrypt** **(config apps inline-ssl profile alias sslprofile) # certificate revocation crl disable** **(config apps inline-ssl profile alias sslprofile) # certificate revocation ocsp disable** **(config apps inline-ssl profile alias sslprofile) # certificate self-signed decrypt** **(config apps inline-ssl profile alias sslprofile) # certificate unknown-ca decrypt** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp inactive-timeout 5** **(config apps inline-ssl profile alias sslprofile) # decrypt tcp portmap default-out-port disable** **(config apps inline-ssl profile alias sslprofile) #**

Step	Description	Command
		decrypt tool-bypass disable **(config apps inline-ssl profile alias sslprofile) # default-action decrypt** **(config apps inline-ssl profile alias sslprofile) # no-decrypt tool-bypass disable** **(config apps inline-ssl profile alias sslprofile) # url-cache miss action decrypt** **(config apps inline-ssl profile alias sslprofile) # exit** **(config) #**
6.	Create a key map entry.	**(config apps inline-ssl profile alias sslprofile) # keymap add server server_ 001.autssl.qa.gigamon.com key server_key_001**
7.	Configure GigaSMART group and associate it with a GigaSMART engine port.	**(config) # gsgroup alias issl1-gsgroup port-list 1/2/e1**
8.	Configure the GigaSMART inline SSL operation, specify the profile, and assign the GigaSMART operation to the GigaSMART group.	**(config) # gsop alias issl1-gsop inline-ssl issl1_prof port-list issl1-gsgroup**
9.	Configure a virtual port.	**(config) # vport alias issl1-vport gsgroup issl1-gsgroup**
10.	Enable HSM group in gsparams.	**(config) # gsparams gsgroup issl1-gsgroup hsm-group add hsm-set**
11.	Configure source and destination ports and enable them.	**(config) # port 1/1/x1 type inline-network** **(config) # port 1/1/x1 params admin enable** **(config) # port 1/1/x9 type inline-tool** **(config) # port 1/1/x9 params admin enable**
12.	Configure inline network.	**(config) # inline-network alias issl1-inline-network** **(config) # pair net-a 1/1/x1 and net-b 1/1/x2** **(config) # physical-bypass disable** **(config) # traffic-path to-inline-tool** **(config) # exit**
13.	Configure inline tool and enable it.	**(config) # inline-tool alias issl1-inline-tool** **(config) # pair tool-a 1/1/x9 and tool-b 1/1/x10** **(config) # enable** **(config) # failover-action tool-bypass**

Step	Description	Command
		(config) # shared true **(config) # hb-profile default** **(config) # heart-beat** **(config) # hb-ip-addr-a 0.0.0.0** **(config) # hb-ip-addr-b 0.0.0.0** **(config) # exit**
14.	Configure map to replace roles from admin to owner	**(config) # map alias issl1_l1_map** **(config map alias issl1_l1_map) # roles replace admin to owner_roles** **(config map alias issl1_l1_map) # rule add pass protocol tcp** **(config map alias issl1_l1_map) # to issl1-vport** **(config map alias issl1_l1_map) # from issl1-inline-network** **(config) # exit** **(config) # map alias issl1_l2_map** **(config map alias issl1_l2_map) # roles replace admin to owner_roles** **(config map alias issl1_l2_map) # use gsop issl1-gsop** **(config map alias issl1_l2_map) # to issl1-inline-tool** **(config map alias issl1_l2_map) # from issl1-vport** **(config) # exit** **(config) # map-scollector alias SCOL** **(config map-scollector alias SCOL) # roles replace admin to owner_roles** **(config map-scollector alias SCOL) # from issl1-inline-network** **(config map-scollector alias SCOL) # collector bypass** **(config) # exit**
15.	Register the configured GigaSMART client in the Luna server.	**Config # show gigasmart engine details** **lunash:>client register -client 10.115.74.195 -ip 10.115.54.168** **lunash:>client assignpartition -client 10.115.74.195 -partition fqa-par1-1** Notes: • Make sure to wait for at least 60 seconds before you register the client on the Luna server. The total maximum wait time is 10 minutes. In case of reload, the minimum wait time is 90 seconds before you proceed with the registration.

Step	Description	Command
		• For HA group configuration, you should register the GigaSMART client in both the Luna servers. • For other administrative commands, refer to Client Register - Luna Command Reference.

Related Commands

Task	Command
Displays the HSM Group status. **Note**: Before registering the client in the Luna server, the operational status will be displayed as **Init**. Once the client registration is complete, the status will be changed to **Yes** after 30 seconds.	(config) # show apps hsm-group status

Display GigaSMART Statistics

Use the following commands to display GigaSMART parameters, operations, and groups:

Command	Summary
show gsgroup alias <alias> all flow-ops-report alias <alias> type flow-sampling | ssl-decryption | flow-filtering <any | device-ip-mask <IP address> <netmask> | gtp-imsi-pattern | imei-pattern | msisdn-pattern> [summary | upload <upload URL>] flow-sip <any | callerid-pattern> [summary | upload <upload URL>] inline-ssl any upload <upload URL> flow-whitelist alias <GTP whitelist file alias> imsi <IMSI number> gsapp-resource <alias <alias> | all> gtp-persistence <alias <alias> | all> sip-fwdlist <alias <alias> caller-id <caller ID>> stats [alias <alias> | all]	Use this command to review settings and statistics for GigaSMART groups. A GigaSMART group is a combination of one or more GigaSMART engine ports available in a single GigaVUE HC Series chassis. GigaSMART engine ports can be combined into groups. GigaSMART engine ports are numbered using **<bid/sid/e1..e2>**; for example, a GigaSMART in box ID 3, slot 2, has GigaSMART engine ports **3/2/e1** and **3/2/e2** available for grouping. Similarly, a GigaVUE-HC1 node with the box ID of 5 has GigaSMART engine port 5/1/e1 (and even though there is only a single GigaSMART engine port in the GigaVUE-HC1 node, it still must be set as a GigaSMART group). • Use the **alias** argument to see the GigaSMART engine ports included in a specific GigaSMART group. • Use the **all** argument to see the GigaSMART engine ports included in each GigaSMART group configured on the node. • Use the **flow-ops-report** argument to see session tables for the following: o flow-sampling—Displays flow aware sampling. o flow-filtering—Displays flow aware filtering. o flow-sip—Fetches a report of SIP/RTP flows. o ssl-decryption—Displays Passive SSL decryption. o inline-ssl—Uploads SSL Decryption for inline tools session log file. **Note**: In a cluster environment, **flow-ops-report** only displays output for the GigaSMART group defined in the local node. (This is due to the large amounts of data stored in the session table.) The following is a sample error message: **Flowops report won't display to terminal since port 5_2_e1 of gsgroup doesn't belong to this box** • Use the **flow-whitelist** argument to display a particular IMSI associated with the GigaSMART group. • Use the **gsapp-resource** argument to display GigaSMART application resource usage. Refer to *Displaying GigaSMART Application Resource Usage* on page 779. • Use the **gtp-persistence** argument to display information for GTP stateful session recovery. Refer to GigaSMART GTP Stateful Session Recovery. • Use the **sip-whitelist** argument to fetch a report of a SIP whitelist. • Use the **stats** argument to see packet and byte counts by GigaSMART group. You can also use the **stats** argument to check for packets dropped by a GigaSMART group. If

Command	Summary
	packets are being dropped, this is an indication that you may have oversubscribed the GigaSMART group with operations. Each GigaVUE-OS-HC0 GigaSMART engine port can process a maximum of 40Gb. Each GigaSMART engine port on the SMT-HC3-C05 module on GigaVUE-HC3 can process packets at up to 100Gb. The GigaVUE-HC1 GigaSMART engine port can process a maximum of 20Gb. You can combine the GigaVUE-HC3 node into a larger group for more processing power, if needed.
show gsparams [alias <alias> | all]	Use this command to check the current setting of GigaSMART parameters for each GigaSMART group in the chassis.
show gsop alias <alias> all by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt> stats [alias <alias> [ip-frag] | all [detail] | by-application <add-header | dedup | apf | asf | flow-sampling | flow-filtering | lb | masking | slicing | strip-header | trailer | tunnel-decap | ssl-decrypt] by-gsgroup <GS group alias>>	Use this command to review statistics for existing GigaSMART operations: • Use the **alias**, **all**, and **by-application** arguments to review the configuration of different GigaSMART operations. • Use the **stats** argument to view packet processing statistics for GigaSMART operations (packets/bytes counts, packet drop counts, duplicates detected, and so on)

GigaSMART Trailers

GigaSMART operations can add the GigaSMART Trailer to packets, providing metadata on where the packet was processed.

GigaSMART Trailers are optional for some GigaSMART operations. Refer to How to Combine GigaSMART Operations for the valid combinations.

Example 1 – GigaSMART Source Port Labeling with a GigaSMART Trailer

Summary	Command
This example creates a GigaSMART operation named **src_headermask** with **masking** and **trailer** components. This operation will mask packets using a static masking offset of 148 bytes that continues for the next 81 bytes, writing over the existing data with an FF pattern. Then it attaches a GigaSMART trailer indicating the original size of the packet before masking, the original packet's CRC, and the box ID, slot ID, and port ID of the physical input port on theGigaVUE HC Series node.	**(config) # gsop alias src_headermask masking protocol none offset 148 pattern 0xFF length 81 trailer add crc enable srcid enable port-list GS1**

Example 2 – GigaSMART Source Port Labeling with a GigaSMART Trailer

Summary	Command
This example creates a GigaSMART operation named **src_headerslice** with **slicing** and **trailer** components. This operation will slice packets 4 bytes after the UDP layer and attach a GigaSMART trailer indicating the original size of the packet before slicing, the original packet's CRC, and the box ID, slot ID, and port ID of the physical input port on the GigaVUE HC Seriesnode.	**(config) # gsop alias src_headerslice slicing protocol udp offset 4 trailer add crc enable srcid enable port-list GS2**

Displaying Trailer Statistics

To display trailer statistics, use the following CLI command:

(config) # show gsop stats alias forTrailer

Remove GigaSMART Trailers

You can also construct GigaSMART operations that remove the GigaSMART Trailer from packets. These operations are useful in cases where you have cascade connections – a tool port receiving packets with a GigaSMART trailer is physically cabled to a GigaVUE HC Series network port, sending the packets received on the tool port back into a GigaVUE HC Series node. You may want to remove the GigaSMART trailer before the packets are forwarded to other tools – that is when the **trailer remove** argument comes in handy.

Configure Clustering

A cluster consists of multiple GigaVUE-OS nodes operating as a unified fabric such that packets entering the cluster on one node can be sent to a destination port on any other node. You set up packet distribution using the standard box ID/slot ID/port ID format, allowing maps to distribute traffic to any port in the cluster. The nodes in a cluster must belong to the same software version.

The configuration examples for clustering is described in the following sections:

- Clustering a Node Using Layer 3 Out-of-Band Manual Discovery
- Configuring Layer 3 Out-of-Band Manual Discovery
- Create and Execute the Configuration Plans
- How to Use Jump-Start Configuration on GigaVUE-OS®TA Series Nodes
- Join a Node to a Cluster (Out-of-Band)

Related Topics

- Refer to the *"GigaVUE-OS Nodes and Clusters"* chapter in the *GigaVUE Fabric Management Guide* for detailed information about clustering.
- Refer to the cluster in the reference section for details of the syntax of the cluster CLI command.

Clustering a Node Using Layer 3 Out-of-Band Manual Discovery

Starting in software version 5.1, a node residing on a different management subnet can join an out-of-band cluster using Layer 3 (L3) out-of-band manual discovery.

A node residing on a different IP subnet manually discovers the IP address of the current leader in the cluster and the IP address of the standby node. After discovering the IP addresses, the node residing on a different subnet establishes the connection with the current leader in the cluster. Once the node joins the cluster, it automatically receives a complete copy of the cluster's database.

To allow a node on a different subnet to manually discover the primary leader IP address and the secondary standby IP address in the cluster, the auto-discovery of the cluster leader must be disabled. When the auto-discovery is disabled, the discovery process is manual. Also, the primary and secondary IP addresses must be configured for manual discovery.

> **NOTE:** The leader and the standby nodes must be reachable by the nodes residing on a different subnet.

For example:

```
(config) # no cluster leader auto-discovery
(config) # cluster leader address primary ip 192.168.1.52 port 60102
```

(config) # cluster leader address secondary ip 192.168.1.54 port 60102

NOTE: When using manual discovery, the virtual IP address (vip) is not supported.

Manual discovery uses the primary IP address. When a leader fails and the standby is promoted to be the new leader, the node uses the secondary IP address and connects to the new leader. The node must discover the new leader within the specified timeout value.

For example:

(config) # cluster leader connect timeout 40

The default is 15 seconds. The values range from 10 to 120 seconds.

The nodes residing on a different subnet are not capable of becoming a leader or a standby node. They can only have the role of a normal node. The cluster leader preference assigned to these nodes are ignored.

Configuring Layer 3 Out-of-Band Manual Discovery

If the Mgmt IP address of all nodes reside on a different subnet and you want to put them into an out-of-band cluster, use the procedure in this section.

Refer to the following configuration example:

Step	Description	Command
1.	Specify the cluster name, cluster ID, and cluster interface for all the nodes that will be part of the cluster. **Note**: Only eth0 interface is used for L3 out-of-band manual discovery.	**(config) # cluster name Layer3-oob** **(config) # cluster id Layer3-oob** **(config) # cluster interface eth0**
2.	Verify that all the nodes have the same cluster information and that they are all running the same software version.	**(config) # show cluster config** **(config) # show version**
3.	Specify the cluster leader preference. For the node that is to be the leader, configure the highest preference, for example, 100. (For member nodes, the preference can be between 10 and 99.) **Note**: This command is not available for GigaVUE-OS TA	**(config) # cluster leader preference 100**

Step	Description	Command
	series.	
4.	Ping the Mgmt IP address of the node. If ping is good, enable the cluster on the leader first and then on the standby node.	**(config) # ping mgmt-ip** **(config) # cluster enable**
5.	Verify that the cluster has formed.	**(config) # show cluster global brief** **(config) # show chassis**
6.	On the leader, add the box ID of the other nodes in the cluster.	**(config) # chassis box-id <box ID> serial-number <serial number>**
7.	For all other nodes residing on the different subnet, disable auto-discovery of the cluster leader. To enable L3 out-of-band manual discovery, configure the IP address used by the leader and the standby node.	**(config) # no cluster leader auto-discovery** **(config) # cluster leader address primary ip <IP address of leader>** **(config) # cluster leader address secondary ip <IP address of standby>**
8.	Verify the configuration on a local node. Confirm the connectivity to the leader and the standby node.	**(config) # show cluster config** **(config) # ping mgmt-ip <IP address of leader>** **(config) # ping mgmt-ip <IP address of standby>**
9.	Once the connectivity is confirmed, enable the cluster on the local node.	**(config) # cluster enable**
10.	On the leader, repeat Step 5 and Step 6. Save the configuration.	**(config) # write mem**
11.	Display cluster configuration	**(config) # show cluster configured**

How to Create a Cluster

Setting up a cluster consists of a number of steps. Refer to the *"Creating Clusters: A Roadmap"* section in the *GigaVUE Fabric Management Guide* for detailed information. For configuration examples, refer to the following sections:

Create and Execute the Configuration Plans

Once you have drawn your cluster topology, it is easy to write up configuration plans for each node in the cluster showing the values for the configuration commands you will need to issue. For example, the plans for the cluster topology in Figure 12-4 could look like those in the following tables.

These plans all use HCCv2 control cards and establish cluster connectivity over the cluster management ports and then go on to set up the stack-links. **You cannot establish stack-links between nodes until the cluster itself is communicating.**

Note that the easiest way to establish a node's cluster settings is with the **config jump-start** script described in the **Hardware Installation Guides**. The script is illustrated as follows. The values entered for Steps 16-21 matching those in the configuration plan for our first node:

```
Gigamon GigaVUE-OS HC Series Chassis
 gigamon-0d04f1 login: admin
Gigamon GigaVUE-OS HC Series Chassis
GigaVUE-OS configuration wizard
 Do you want to use the wizard for initial configuration? yes
 Step 1: Hostname? [gigamon-0d04f1] Node_A
 Step 2: Management interface? [eth0]
 Step 3: Use DHCP on eth0 interface? no
 Step 4: Use zeroconf on eth0 interface? [no]
 Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.2/24
 Step 6: Default gateway? 10.150.52.1
 Step 7: Primary DNS server? 192.168.2.20
 Step 8: Domain name? gigamon.com
 Step 9: Enable IPv6? [yes]
 Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
 Step 15: Admin password?

 Please enter a password. Password is a must.

 Step 15: Admin password?
 Step 15: Confirm admin password?
 Step 16: Cluster enable? [no] yes
 Step 17: Cluster interface? [eth0]
 Step 18: Cluster id (Back-end may take time to proceed)? [default-cluster] 1010
 Step 19: Cluster name? [default-cluster] 1010
Step 20: Cluster Leader Preference (Enter higher leader preference for the node to be elected as leader)?
[65]
 Step 21: Cluster mgmt IP address and masklen? [0.0.0.0/0] 10.150.56.71/24
```

How to Use Jump-Start Configuration on GigaVUE-OS® TA Series Nodes

If a license to enable the cluster is not available when first configuring jump-start, you will see the following output:

TA1 (config) # configuration jump-start

GigaVUE-OS configuration wizard

Step 1: Hostname? [TA1]

Step 2: Management Interface <eth0> ? [eth0]
Step 3: Use DHCP on eth0 interface? [yes]
Step 4: Enable IPv6? [yes]
Step 5: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 6: Enable DHCPv6 on eth0 interface? [no]
Step 7: Enable secure cryptography? [no]
Step 8: Enable secure passwords? [no]
Step 9: Minimum password length? [8]
Step 10: Admin password?

Please enter a password. Password is a must.

Step 10: Admin password?
Step 10: Confirm admin password?

No valid advanced features license found!

You have entered the following information:

1. Hostname: TA1
2. Management Interface <eth0> : eth0
3. Use DHCP on eth0 interface: yes
4. Enable IPv6: yes
5. Enable IPv6 autoconfig (SLAAC) on eth0 interface: no
6. Enable DHCPv6 on eth0 interface: no
7. Enable secure cryptography: no
8. Enable secure passwords: no
9. Minimum password length: 8
10. Admin password: ********

To change an answer, enter the step number to return to.
Otherwise hit <enter> to save changes and exit.

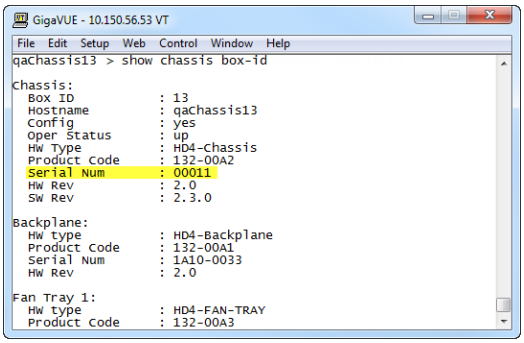
Choice:

Configuration changes saved

Once the **license is installed**, you can run jump-start again and the steps relating to enabling the cluster will become available, as follows:

Step 11: Cluster enable? [yes]
Step 12: Cluster Interface <eth0> ? [eth0]
Step 13: Cluster id (Back-end may take time to proceed)? [89]
Step 14: Cluster name? [Cluster-89]
Step 15: Cluster mgmt IP address and masklen? [10.115.25.89/21]

Configure Cluster Connectivity – Configuration Plans

Configuration Plan for qaChassis 13 (Box ID 13)		Commands
Cluster ID	1010	**cluster id 1010**
Cluster Name	1010	**cluster name 1010**
Cluster Leader VIP	10.150.56.71 /24	**cluster leader address vip 10.150.56.71 /24**
Cluster Control Interface	eth0	**cluster interface eth0**
Cluster Mgmt Port IP (eth0)	zeroconf	**interface eth0 zeroconf(IP Configuration for eth0 obtained automatically through default zeroconf setting)**
Enable Clustering	Yes	**cluster enable**
Record Serial Number	00011	**show chassis** output: 
Box ID	13	**chassis box-id 13 serial-num 00011**
Because this is the first node we are configuring with this cluster ID and leader VIP, it automatically assumes the leader role. Configure cluster connectivity for the other nodes before assigning their box IDs to their serial numbers and configuring stack-links from the leader VIP address.		

Configuration Plan for qaChassis 14 (Box ID 14)		Commands
Cluster ID	1010	**cluster id 1010**
Cluster Name	1010	**cluster name 1010**
Cluster Leader VIP	10.150.56.71 /24	**cluster leader address vip 10.150.56.71 /24**
Cluster Control Interface	eth0	**cluster interface eth0**
Cluster Mgmt Port IP (eth0)	zeroconf	**interface eth0 zeroconf(IP Configuration for eth0 obtained automatically through default**

Configuration Plan for qaChassis 14 (Box ID 14)		Commands
		zeroconf setting)
Enable Clustering	Yes	**cluster enable**
Record Chassis Serial Number You will need the chassis serial number when you add this node's box ID to the leader's database later on.	80052	show chassis
Box ID	14	on page 618).

Configuration Plan for qaChassis 10 (Box ID 10)		Commands
Cluster ID	1010	**cluster id 1010**
Cluster Name	1010	**cluster name 1010**
Cluster Leader VIP	10.150.56.71 /24	**cluster leader address vip 10.150.56.71 /24**
Cluster Control Interface	eth0	**cluster interface eth0**
Cluster Mgmt Port IP (eth0)	zeroconf	**interface eth0 zeroconf (IP Configuration for eth0 obtained automatically through default zeroconf setting)**
Enable Clustering	Yes	**cluster enable**
Record Chassis Serial Number You will need the chassis serial number when you add this node's box ID to the leader's database later on.	80054	show chassis
Box ID	10	**(assigned to normal node from leader using "chassis box-id <box ID> serial-num <serial number>" after cluster connectivity is established. Refer to *Connect to the Leader and Add the Member Nodes to the Database***

Configuration Plan for qaChassis 11 (Box ID 11)		Commands
Cluster ID	1010	**cluster id 1010**

Configuration Plan for qaChassis 11 (Box ID 11)		Commands
Cluster Name	1010	**cluster name 1010**
Cluster Leader VIP	10.150.56.71 /24	**cluster leader address vip 10.150.56.71 /24**
Cluster Control Interface	eth0	**cluster interface eth0**
Cluster Mgmt Port IP (eth0)	zeroconf	**interface eth0 zeroconf(IP Configuration for eth0 obtained automatically through default zeroconf setting)**
Enable Clustering	Yes	**cluster enable**
Record Chassis Serial Number You will need the chassis serial number when you add this node's box ID to the leader's database later on.	00007	show chassis
Box ID	11	(assigned to normal node from leader using "chassis box-id <box ID> serial-num <serial number>" after cluster connectivity is established. Refer to *Connect to the Leader and Add the Member Nodes to the Database*.

Connect to the Leader and Add the Member Nodes to the Database

Once you have made the configuration settings necessary to establish cluster connectivity, you need to register each normal node with the leader so that their box IDs and card configuration are in its database. You do this with the **chassis box-id <box ID> serial-num <serial number>** and **card all box-id <box ID>** commands in the leader VIP CLI.

Use the following procedure:

1. Log in to the leader VIP CLI. This example uses 10.150.56.71 for the leader VIP address.
2. For each normal node in the cluster, we need to add the box ID and card configuration. Use the following commands:

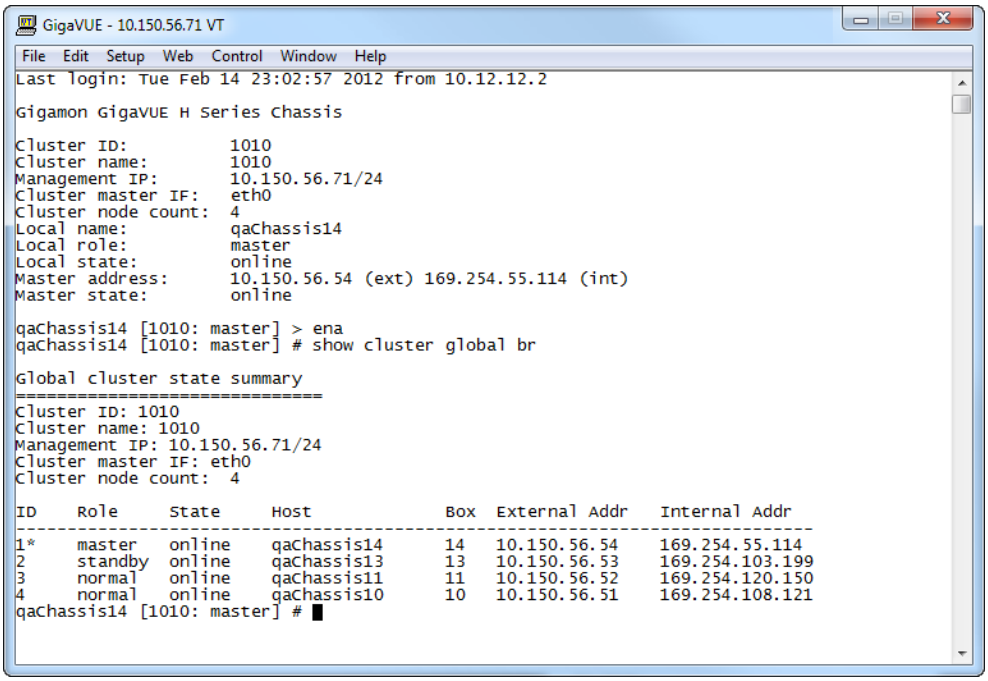
Box ID	Description	Commands
14	Registers the chassis with the serial number of 80052 as box ID 14 and adds all its cards to the database.	**chassis box-id 14 serial-num 80052** **card all box-id 14**
10	Registers the chassis with the serial number of 80054	**chassis box-id 10 serial-num 80054**

Box ID	Description	Commands
	as box ID 10 and adds all its cards to the database.	**card all box-id 10**
11	Registers the chassis with the serial number of 00007 as box ID 11 and adds all its cards to the database.	**chassis box-id 11 serial-num 00007** **card all box-id 11**
Note: The box IDs you specify here do not need to match the ones you set up with **config jump-start**. The leader applies the box ID to the specified chassis serial number with the commands here. However, for the sake of consistency and ease of configuration, it is generally easiest to use matching box IDs.		

It can take a minute or two for the **card all box-id** command to complete. Once it does for all nodes in the cluster, the **show cluster global** command displays the box IDs you configured in the table. Refer to the next section for an example.

Verify Cluster Connectivity

Once you have made the configuration settings necessary to establish cluster connectivity, verify that the connections you expect are there by connecting to the leader VIP address and using the **show cluster global brief** command. For example:



From here, we can see that all four nodes are connected to the cluster. Each node has an External and an Internal Address as follows:

- **External Address** – The IP address assigned to the Mgmt port on the control card for each node. When working with a cluster configuration, this IP address is active for CLI and GUI management of the local node. However, cluster-wide tasks should be performed using the virtual IP address for the cluster. This virtual IP address is assigned to whichever node is currently performing the leader role. Should the leader go down, causing the standby node to be promoted to leader, the new leader will take over the virtual IP address for the cluster.

  NOTE: Active connections to the leader VIP will be dropped when a new node is promoted to the leader role and takes ownership of the address. You can reconnect to the leader VIP to resume operations.

- **Internal Address** – The IP address assigned to the cluster management port on the control card for this node. This address is used for cluster management traffic. It is referred to as internal because you never work over this interface directly – the cluster uses it for stack management.

Configure Stack-Links from Leader

Once you have verified that the cluster is successfully communicating, you can connect to the leader VIP, enable the ports for the stack-links, and finally configure the stack-links. Use the **port <port-list> params admin enable** command to enable ports. Then, per our cluster topology, configure the following stack-links:

Stack Links for Cluster 1010	Ports	Commands
qaChassis13 to qaChassis14	13/1/x8 to 14/8/x8	First, set the port-type to stack for both ends of the stack-link. Then, connect them with the stack-link command: **port 13/1/x8 type stack** port 14/8/x8 type stack **stack-link alias c13-to-c14 between ports 13/1/x8 and 14/8/x8**
qaChassis13 to qaChassis10	13/1/x9..x12 to 10/2/x1..x4	First, configure the GigaStream on both sides of the stack-link. Then, connect them with the stack-link command: **port 13/1/x9..x12 type stack** **gigastream alias 13stack port-list 13/1/x9..x12** **port 10/2/x1..x4 type stack** gigastream alias 10stack port-list 10/2/x1..x4 **stack-link alias c13-to-c10 between gigastreams 13stack and 10stack**
qaChassis10 to qaChassis11	10/1/x12 to 11/1/x12	**stack-link alias c10-to-c11 between ports 10/1/x12 and 11/1/x12**

Once you have configured your stack-links, check them with the **show-stack-link** command to make sure they are up. Figure 4-25 provides an example of the **show stack-link** output for our sample cluster.

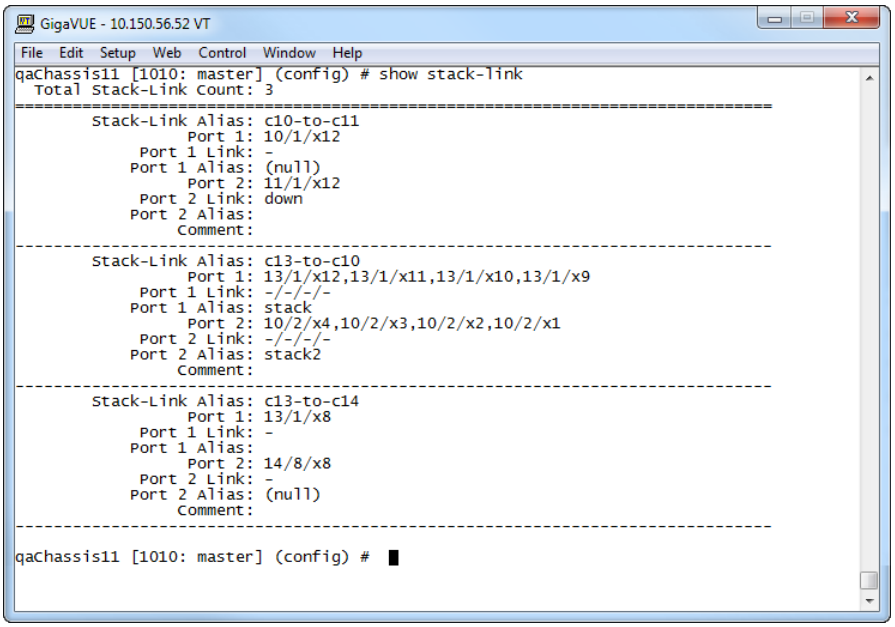
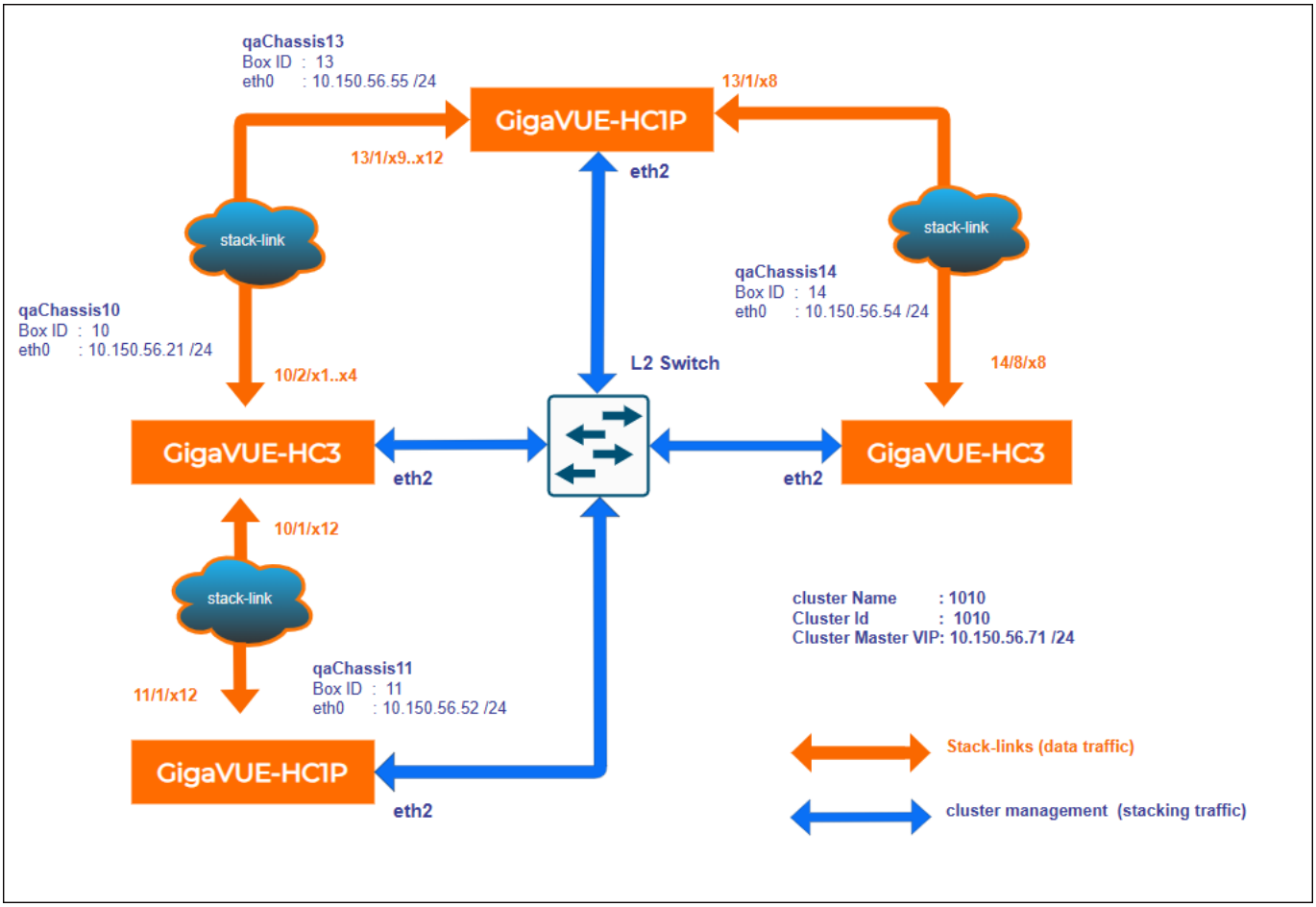


Figure 26 *Output of show stack-link for Sample Cluster*

The cluster is now up and running. You can log into the leader VIP and configure cross-node packet distribution using standard box ID/slot ID/port ID nomenclature. The following figure illustrates the cluster, along with its configuration.



Join a Node to a Cluster (Out-of-Band)

The simplest way to join a node to a cluster is to modify the cluster ID. Refer to Step 3 in the following procedure. However, a good practice for joining a node to a cluster is to include the following commands:

```
(config) # no cluster enable
(config) # no traffic all
(config) # cluster id <cluster ID>
(config) # cluster name <name>
(config) # cluster leader address vip <IP address> <netmask | mask length>
(config) # cluster interface <eth>
(config) # cluster enable
```

NOTE: In Step 5, the syntax includes a space between the IP address and the netmask.

Add a Node to an Existing Cluster – Reset to Factory Defaults

Gigamon recommends resetting a GigaVUE-OS node to its factory settings before adding it to an existing cluster.

If moving a node from one cluster to another, you should use the "reset factory all" command to remove any cluster data in the configs.

To reset factory default settings:

1. Save and upload all configuration files for the node that you want to keep to external storage, including the running configuration. Back up the running configuration using either of the following methods:

Back Up by Copying and Pasting	**(config) # show running-config** This command displays the commands necessary to recreate the node's running configuration on the terminal display. You can copy and paste the output from this command into a text file and save it on your client system. The file can later be pasted back into the CLI to restore the configuration.
Back up to SCP/TFTP/HTTP Server	**(config) # configuration text generate active running upload <upload URL> <filename>** This command uses FTP, TFTP, or SCP to upload the running configuration to a text file on remote storage. The format for the **<upload URL>** is as follows: **[protocol]://username [:password]@hostname/path/filename** For example, the following command uploads a text configuration file based on the active running configuration and uploads it to an FTP server at 192.168.1.49 with the name **config.txt**: **(config) # configuration text generate active running uploadftp://myuser:mypass@192.168.1.49/ftp/config.txt**

2. Run the following command to reset the node to its factory defaults:

   (config) # reset factory only-traffic

   The node reloads automatically.
3. Connect the ports to be used for cluster management traffic.
4. Run the jump-start script with the following command if it does not appear automatically:

   (config) # config jump-start
5. Follow the jump-start script's prompts to configure the node, including the cluster settings for the cluster you want to join.

The reason for this is that both the standalone node and the target cluster have their own name-spaces for traffic-related aliases - maps, tool-mirrors, and so on. When a new node joins a cluster, the system attempts to merge the aliases. However, if there are any duplicate

aliases between the standalone node and the existing cluster, there can be destabilizing results for the newly added node, up to and including system exceptions. Because of this, Gigamon recommends resetting a node's settings with the **reset factory only-traffic** command before adding it to an existing cluster.

Remove a Node from a Cluster and Using as a Standalone

To remove a GigaVUE-OS node from an existing cluster and apply a saved standalone configuration file, perform the following steps:

1. Remove the node from the cluster by executing the following commands:

   (config) # no chassis box-id <node_boxid_to_be_removed>

   (config) #cluster remove <node-id_to_be_removed>
2. Disable the cluster configuration from the node removed in step 1. Use the **no cluster enable** command.
3. Run the following command to reset the node to its factory defaults:

   (config) # reset factory only-traffic

   The node reloads automatically.
4. Apply the saved standalone configuration file.

Inband Cluster Management

Inband Cluster Management simplifies traditional network management and maintenance by creating a virtual device to manage multiple physical nodes. This simplified approach makes it possible to oversee large networks by defining policies that span across multiple devices. The Inband Cluster Management feature is designed to reduce operational cost and extend coverage by eliminating a dedicated management network.

Inband Cluster Management is supported on all GigaVUE-OS nodes.

Inband Cluster Management Pre-Configuration

A dedicated VLAN is reserved for control management traffic traveling through the interface. When a cluster is configured to use the Inband interface, all cluster management packets are sent and received using this interface.

```
(config) # show interface inband
Interface inband status:
   Comment:
   Admin up:           yes
   Link up:            yes
   DHCP running:       no
   IP address:
   Netmask:
.
.
```

.

NOTE: To use the Inband interface, the IP address should be configured statically or using zeroconf. The Inband interfaces on all nodes in the cluster should be contained within the same subnet.

Inband Cluster Management CLI Syntax

The following example shows the general CLI syntax. For detailed information on leader and member node configuration, refer to How to Setup Inband Cluster Management on a New Cluster.

- To check the information about the Inband interface:
  show interface inband

- Use Inband as a cluster interface:
  cluster interface inbandinterface inband zeroconf

OR

cluster interface inbandinterface inband ip address <ip address> /<subnet mask>

- De-configure the Inband cluster using the **no** CLI command:
  no interface [eth2 | eth1] zeroconf
- Configure offline chassis with the **type** parameter:

chassis box-id <box ID> serial-num <serial-number> type [hc3 | ly2r | hc1 | itac | tacx | ta200|ta25|ta25a]

Configure offline line card with the **product-code** parameter:

card slot <box ID>/<slot ID> product-code <product code>

NOTE: The **product-code** parameter is not displayed.

Offline Remote Configuration for GigaVUE-OS® TA Series Nodes

For GigaVUE TA Series nodes, an additional parameter is required on the leader for offline remote configuration.

- Configure offline mode with the **mode** parameter specifically for the GigaVUE-OS-TA1 and Certified Traffic Aggregation White Box, as follows:
  card slot <box ID>/<slot ID> product-code <product code> mode <48x | 56x | 64x>

NOTE: If the mode is not selected for the GigaVUE TA Series node, the default is 48x.

- Configure offline mode with the **mode** parameter specifically for GigaVUE-OS-TA40, as follows:

card slot <box ID>/<slot ID> product-code <product code> **mode** <0x | 16x>

NOTE: If the mode is not selected for the GigaVUE-OS-TA40, the default is 0x.

Refer also to *Setting up Inband Cluster Management with GigaVUE-OS® TA Series (Including a White Box)* on page 868.

Inband Cluster Management Configuration Examples

Inband Cluster Management is used in the following scenarios:

- How to Setup Inband Cluster Management on a New Cluster
- *Setting up Inband Cluster Management with GigaVUE-OS® TA Series (Including a White Box)* on page 868
- How to Switch from Inband Cluster Management to Out-of-Band
- *Switching from Out-of-Band to Inband Cluster Management* on page 889

Configuration Issues to Consider

Before you begin the Inband Cluster Management configuration, it is highly recommended that you understand and adhere to some known configuration issues that need consideration.

NOTE: Ensure that there is a physical connection between the stack ports of the two nodes that are being added to the Inband cluster.

Issue Number	Configuration Issue Description	Workaround
1	If you disable eth2 (HCCv2), GigaSMART will not come up.	Re-enable the interface that you had disabled, eth2 (HCCv2). Then disable the zeroconf feature on the interface.
2	If you change the stack port configuration or reload the node without disabling the cluster, the nodes in the Inband cluster will go to an “unknown” state temporarily. The cluster status of the nodes (normal, standby) will be determined after a few minutes.	First, disable cluster mode on the affected nodes, then apply the new configuration to the leader node. Second, apply the same configuration to the target node and re-enable cluster modes for such nodes.
3	Never delete stack-gigastream with multiple stack ports (admin enabled) because if you remove the GigaStream they become multiple links and can create loops.	To add more stack port(s) to the existing stack-gigastream perform the following: 1) Disable the target port 2) Change the type of the target port to stack 3) Add this port to the existing stack-gigastream 4) Enable the target port

Issue Number	Configuration Issue Description	Workaround
4	It might take some time to populate a large configuration to the leader node during initial deployment.	Disable the cluster on all member nodes using **no cluster enable** command, and then start populating the large configuration to the leader. Once completed, turn on the cluster on all member nodes to allow database synchronization.
5	The **no traffic all** command removes stack-links when Inband cluster management is in use. This results in a situation in which all nodes become a leader.	Use the **keep-stack** argument of the **no traffic all** command, for example: **(config) # no traffic all keep-stack** This command deletes all traffic configuration, resets the port types, but keeps the stack configuration, including stack ports and GigaStream.
6	You cannot apply saved running configuration text to reconfigure the Inband Cluster Management due to the missing "*type*" field in the chassis command.	Manually configure all the nodes in the cluster before using the remainder of the saved running configuration text.

How to Setup Inband Cluster Management on a New Cluster

This example illustrates how to configure a four-node Inband cluster with zeroconf feature enabled. This example covers GigaVUE HC Series nodes in a cluster. To add a GigaVUE TA Series node or a Certified Traffic Aggregation White Box, refer to *Setting up Inband Cluster Management with GigaVUE-OS® TA Series (Including a White Box)* on page 868.

Before you start, identify the node that will be the leader. Also identify the nodes that will be targeted as standby within the cluster.

NOTE: GigaVUE TA Series nodes and white boxes with GigaVUE-OS can only be configured as normal nodes.

In this example, Seattle is the leader.

The nodes to be configured in the Inband cluster are:

Node Number	Node Name	Node Type
1	Seattle	GigaVUE-HC3
2	Washington	GigaVUE-HC3
3	Boston	GigaVUE-HC1 **Note**: The control card is embedded.

To configure the Inband Cluster Management, you must maintain a command shell for the leader as well as target nodes due to the offline configuration that needs to be applied to leader.

Configuration Steps for Leader: Seattle

1. Open an SSH or terminal session to the Seattle node.

Part 1: Using the Jump-Start Wizard to Configure Node 1

2. In config, enter `configuration jump-start` to start the jump-start wizard:

   gigamon-0d0024 > enable
   gigamon-0d0024 # configure terminal
   gigamon-0d0024 (config) # configuration jump-start
   GigaVUE-OS configuration wizard

3. Enter the parameter values to configure the leader.

   Step 1: Hostname? [gigamon-0d0024] Seattle
   Step 2: Management interface <eth0 eth2 eth3>? [eth0]
   Step 3: Use DHCP on eth0 interface? no
   Step 4: Use zeroconf on eth0 interface? [no]
   Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.6/24
   Step 6: Default gateway? 10.150.52.1
   Step 7: Primary DNS server? 192.168.2.20
   Step 8: Domain name? gigamon.com
   Step 9: Enable IPv6? [yes]
   Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
   Step 11: Enable DHCPv6 on eth0 interface? [no]
   Step 12: Enable secure cryptography? [no]
   Step 13: Enable secure passwords? [no]
   Step 14: Minimum password length? [8]
   Step 15: Admin password?
   Please enter a password. Password is a must.
   Step 15: Admin password?
   Step 15: Confirm admin password?

> **NOTE:** In Step 16, **accept the default of No** so that you do not enable the cluster.

Step 16: Cluster enable? [no]

> **NOTE:** In Step 17, assign the box ID of your chassis.

Step 17: Box-id for the chassis? [1] 7

> **NOTE:** To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

Choice:
Configuration changes saved.
System in classic mode
Seattle (config) #

Part 2: Configuring Inband Cluster on the Leader

4. You need to disable the zeroconf feature on the default cluster interface on eth1 of the control card (HCCv2) in the Seattle node, and make cluster interface Inband with relevant cluster information.

   Seattle (config) # no interface eth1 zeroconf
   Seattle (config) # cluster interface inband
   Seattle (config) # cluster id 600
   Seattle (config) # cluster name 600
   Seattle (config) # cluster leader address vip 10.150.52.233 /24
   Seattle (config) # interface inband zeroconf
   Seattle (config) #

5. Enter `show interfaces` to perform a confirmation check.
6. Make sure that no IP address is assigned on eth1 and new IP address is auto assigned for Inband interface.

   Seattle (config) # show interfaces
   .
   .
   .
   Interface eth1 status:
   Comment:
   Admin up: yes
   Link up: yes
   DHCP running: no
   IP address:

   > **NOTE:** The IP address field on eth1 should be empty.

   Netmask:
   IPv6 enabled: no

Speed: 1000Mb/s (auto)
.
.
.
Interface inband status:
Comment:
Admin up: yes
Link up: yes
DHCP running: no
IP address: 169.254.51.255

> **NOTE:** The IP address field is automatically assigned.

Netmask: 255.255.0.0
Seattle (config) #

7. Enter **`show cluster configured`** to display the current cluster configuration.

   Seattle [600: leader] (config) # show cluster configured
   Global cluster config:
   Cluster enabled: no
   Cluster ID: 600
   Cluster name: 600
   Cluster control interface: inband

   > **NOTE:** The cluster control interface is set to Inband.

   Cluster port: 60102
   Cluster expected nodes: 2
   Cluster startup time: 180
   Cluster shared secret: 1234567890123456
   Cluster leader preference: 60
   Cluster leader auto-discovery enabled: yes
   Cluster leader manual port: 60102
   Cluster leader virtual IP address: 10.150.52.233/24
   Cluster leader management interface: eth0
   Seattle [600:leader] (config) #

Part 3: The Configured Leader is Ready for Inband Cluster

8. Enter the card slot and number command. If the designated stack port is located at slot 4, then wait for the card at slot 4 to come “up” to the “oper state.”

   Seattle (config) # card slot 7/4

9. Enter `show card` to perform a confirmation check.

   Seattle [600: leader] (config) # show card

10. Assign and enable the stack ports. Make them as a stack GigaStream.
11. Enable the cluster.

> **NOTE:** The Seattle node is now the leader as indicated in the CLI prompt.

Seattle (config) # port 7/4/x5..x20 type stack
Seattle (config) # port 7/4/x5..x20 params admin enable
Seattle (config) # gigastream alias big_bridge_7to4 port 7/4/x5..x20
Seattle (config) # cluster enable
Seattle [600: leader] (config) #

Part 4: Apply Offline Remote Node Configuration on the Leader Node

12. Apply offline remote node configuration on the leader node as shown in the CLI command to conclude leader setup.

Seattle [600: leader] (config) # chassis box-id 8 serial-num 12340 type hc3
Seattle [600: leader] (config) # card slot 8/1 product-code 132-0087
Seattle [600: leader] (config) # port 8/1/x5..x20 type stack
! Box '8' is down, unable to validate SFP type for stack port.

> **NOTE:** The `box number is down, unable to validate SFP type for stack port` message is expected behavior.

.
.
.
Seattle [600: leader] (config) # port 8/1/x5..x20 params admin enable
Seattle [600: leader] (config) # gigastream alias big_bridge_8to7 port 8/1/x5..x20
Seattle [600: leader] (config) # write memory

1Enter `show running-config` to perform a confirmation check.

Seattle [600: leader] (config) # show running-config

Configuration Steps for Standby Node: Washington

1. Open an SSH or terminal session to the Washington node.

Part 1: Using the Jump-Start Wizard to Configure Node 2

2. In the command shell for the Washington node, enter the following commands to start the jump-start wizard:
   - enable
   - configure terminal
   - configuration jump-start

Gigamon GigaVUE-OS Chassis
System in classic mode
gigamon-040077 > enable
gigamon-040077 # configure terminal
gigamon-040077 (config) # configuration jump-start

GigaVUE-OS configuration wizard

2Enter the parameter values to configure the standby node.

Step 1: Hostname? [gigamon-040077] Washington
Step 2: Management interface <eth0 eth2 eth3>? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no] no
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.8/24
Step 6: Default gateway? 10.150.52.1
Step 7: Primary DNS server? 192.168.2.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes] yes
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no] no
Step 11: Enable DHCPv6 on eth0 interface? [no] no
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password)?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?

> **NOTE:** In Step 16, accept the default of **No** so that you do not enable the cluster.

Step 16: Cluster enable? [no] no

> **NOTE:** In Step 17, the value 8 indicates the box ID that you assign. Assign your box ID.

Step 17: Box-id for the chassis? [1] 8

> **NOTE:** To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

Choice:
Configuration changes saved.
System in classic mode

Part 2: Configure Inband Cluster on the Remote Target Node 2

3. You need to disable the zeroconf feature on the default cluster interface on eth1 of the control card (HCCv2) in the Washington node and make cluster interface Inband with relevant cluster information.
4. Enter the parameter values to disable the zeroconf feature as shown in the CLI example.

   Washington (config) # no interface eth1 zeroconf

> **NOTE:** The zeroconf is disabled on the default cluster interface of HCCv2 (eth1).

```
Washington (config) # cluster interface inband
Washington (config) # cluster id 600
Washington (config) # cluster name 600
Washington (config) # cluster leader address vip 10.150.52.233 /24
Washington (config) # interface inband zeroconf
Washington (config) # card slot 8/1
Washington (config) # port 8/1/x5..x20 type stack
Washington (config) # port 8/1/x5..x20 params admin enable
Washington (config) # gigastream alias big_bridge_8to7 port 8/1/x5..x20
Washington (config) # wr mem
```

5. Enter **`show interfaces`** to perform a confirmation check.
6. Make sure that there is no IP address assigned for eth1 and that IP address is auto assigned for Inband interface.

```
Washington (config) # show interfaces
.
.
.
Interface eth1 status:
Comment:
Admin up: yes
Link up: yes
DHCP running: no
IP address:
```

> **NOTE:** The IP address field is NULL for eth1.

```
Netmask:
IPv6 enabled: no
.
.
.
Interface inband status:
Comment:
Admin up: yes
Link up: yes
DHCP running: no
IP address: 169.254.228.191
Netmask: 255.255.0.0
IPv6 enabled: yes
.
.
```

.
Washington (config) #

7. Enter **`show cluster configured`** to display the cluster configuration settings.
8. Make sure that the Inband value is defined in the cluster control interface.

Washington (config) # show cluster configured
Global cluster config:
Cluster enabled: no
Cluster ID: 600
Cluster name: 600
Cluster control interface: inband

> **NOTE:** The cluster control interface is set to Inband.

Cluster port: 60102
Cluster expected nodes: 1
Cluster startup time: 180
Cluster shared secret: 1234567890123456
Cluster leader preference: 50
Cluster leader auto-discovery enabled: yes
Cluster leader manual port: 60102
Cluster leader virtual IP address: 10.150.52.233/24
Cluster leader management interface: eth0
Washington (config) #
Washington (config) # show port params port 8/1/x5
Parameter 8/1/x5
====================== ===============
Name Alias:
Type: stack
Admin: enabled
Link status: up

> **NOTE:** The Link Status indicates that the stack port is “up” state.

Auto Negotiate: off
Duplex: full
Speed (Mbps): 10000
MTU: 9400
Force Link Up: off
...

9. Ping the Washington Inband interface.

Seattle [600: leader] (config) # ping 169.254.228.191
PING 169.254.228.191 (169.254.228.191) 56(84) bytes of data.
64 bytes from 169.254.228.191: icmp_seq=1 ttl=64 time=2.10 ms
64 bytes from 169.254.228.191: icmp_seq=2 ttl=64 time=0.153 ms

64 bytes from 169.254.228.191: icmp_seq=3 ttl=64 time=0.145 ms
64 bytes from 169.254.228.191: icmp_seq=4 ttl=64 time=0.135 ms

Part 3: Enable the Cluster or Remote Target Node

10. Configure the cluster role of Node 2 to be “standby”.

    Washington (config) # cluster enable
    Washington [600: unknown] (config) #

    > **NOTE:** The Washington node is in an unknown transitional state.

    Washington [600: standby] (config) #

    > **NOTE:** The Washington node is now a standby and the joining with the leader is complete.

11. On the leader command shell, enter `show chassis` to perform a confirmation check if the chassis are in the “up” and “oper status”.

    Seattle [600: leader] (config) # show chassis
    Box# Hostname Config Oper Status HW Type Product# Serial# HW Rev SW Rev
    ---
    7 * Seattle yes up HC3-Chassis 132-0098 80016 A0 3.2.00
    8 Washington yes up HC3-Chassis 132-0098 12340 AA 3.2.00
    Seattle [600: leader] (config) #

Configuration Steps for Node 3: Boston

You will test the third node joining action. For two nodes Inband cluster setup, apply all configuration values including the joining node on the leader.

You will then configure the remote target node. For the third node to join, the leader must already have the second node configuration information. You need to preserve this portion of the configuration information on the leader. Therefore append the additional third node configuration on the top of the existing information.

1. Open an SSH or terminal session to the Boston node.

Part 1: Using the Jump-Start Wizard to Configure Node 3

2. In the command shell for the Washington node, enter the following commands to start the jump-start wizard:
    - enable
    - configure terminal
    - configuration jump-start

    Gigamon GigaVUE-OS Chassis
    System in classic mode
    gigamon-0d0025 > enable
    gigamon-0d0025 # configure terminal

gigamon-0d0025 (config) # configuration jump-start

3. Enter the parameter values to configure the target node.

GigaVUE-OS configuration wizard
Do you want to use the wizard for initial configuration? yes
Step 1: Hostname? [gigamon-0d0025] Boston
Step 2: Management interface? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no]
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.20/24
Step 6: Default gateway? 10.150.52.1
Step 7: Primary DNS server? 192.168.2.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?

> **NOTE:** In Step 16, accept the default **No**.

Step 16: Cluster enable? [no]

> **NOTE:** In Step 17, assign a box ID for node 3.

Step 17: Box-id for the chassis? [1] 21

> **NOTE:** To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

Choice:
Configuration changes saved.
To return to the wizard from the CLI, enter the "configuration jump-start" command from configure mode. Launching CLI...
System in classic mode
Boston > enable
Boston # configure terminal
Boston (config) #

Part 2: Configure Inband Cluster on the Remote Target Node 3

4. You need to disable the zeroconf feature on the default cluster interface on eth1 of the control card (HCCv2) in the Boston node and make cluster interface Inband with relevant cluster information.
5. Enter the parameter values to disable the zeroconf feature as shown in the CLI example.

   Boston (config) # no interface eth1 zeroconf

   > **NOTE:** The zeroconf is disabled on eth1.

   Boston (config) # cluster interface inband
   Boston (config) # cluster id 600
   Boston (config) # cluster name 600
   Boston (config) # cluster leader address vip 10.150.52.233 /24
   Boston (config) # interface inband zeroconf
6. Enter **`show interfaces`** to perform a confirmation check.
7. Make sure that there is no IP address assigned for eth1 and that IP address is auto assigned for Inband interface.
8. Make sure that the cluster control interface displays Inband value.

   Boston (config) # show interfaces
   .
   .
   .
   Interface eth1 status:
   Comment:
   Admin up: yes
   Link up: yes
   DHCP running: no
   IP address:

   > **NOTE:** The IP address field is NULL for eth1.

   Netmask:
   IPv6 enabled: no
   .
   .
   .
   Interface inband status:
   Comment:
   Admin up: yes
   Link up: yes
   DHCP running: no
   IP address: 169.254.145.136

> **NOTE:** The IP address field is automatically assigned.

Netmask: 255.255.0.0
IPv6 enabled: yes
.
.
.
Boston (config) #
Boston (config) # show cluster configured
Global cluster config:
Cluster enabled: no
Cluster ID: 600
Cluster name: 600
Cluster control interface: inband <-- inband

> **NOTE:** The cluster control interface is set to Inband.

Cluster port: 60102
Cluster expected nodes: 1
Cluster startup time: 180
Cluster shared secret: 1234567890123456
Cluster leader preference: 60
Cluster leader auto-discovery enabled: yes
Cluster leader manual port: 60102
Cluster leader virtual IP address: 10.150.52.233/24
Cluster leader management interface: eth0
Boston (config) #

Part 3: Configure Relevant Stack Ports and Node 3 Configuration on the Leader

9. On the leader command shell, configure local stack ports on the leader. Enter the configuration information as shown.

Seattle [600: leader] (config) # card slot 4/2
Seattle [600: leader] (config) # port 4/2/x5..x6 type stack
Seattle [600: leader] (config) # port 4/2/x5..x6 params admin enable
Seattle [600: leader] (config) # gigastream alias smaller_bridge_4to21 port 4/2/x5..x6

3Configure offline stack port for Node 3.

Seattle [600: leader] (config) # chassis box-id 21 serial-num 40263 type hc3
Seattle [600: leader] (config) # card slot 21/3 product-code 132-0045
Seattle [600: leader] (config) # port 21/3/x1..x2 type stack
! Box '21' is down, unable to validate SFP type for stack port.
Seattle [600: leader] (config) # port 21/3/x1..x2 params admin enable
Seattle [600: leader] (config) # gigastream alias smaller_bridge_21to7 port 21/3/x1..x2

10. Enter **`show running-config`** to perform a confirmation check.

    Seattle [600: leader] (config) # show running-config
    ##

Part 4: Configure Stack Ports for Joining Node 3

11. In the command shell for Node 3, enter the stack port configuration information.

    Boston (config) # card slot 21/3
    Boston (config) # port 21/3/x1..x2 type stack
    Boston (config) # port 21/3/x1..x2 params admin enable
    Boston (config) # gigastream alias smaller_bridge_21to7 port 21/3/x1..x2

12. Enter `show port params` to perform a confirmation check.

    Boston (config) # show port params port 21/3/x1..x2
    Parameter 21/3/x1 21/3/x2
    ====================== ============== ===============
    Name Alias:
    Type: stack stack

    > **NOTE:** The stack values indicates the state of the port.

    Admin: enabled enabled
    Link status: up up

    > **NOTE:** The Link Status indicates the port's status. In this case, it is "up".

    Auto Negotiate: off off
    Duplex: full full
    Speed (Mbps): 10000 10000
    MTU: 9400 9600
    Force Link Up: off off
    Port Relay: N/A N/A
    ...

    4On the command shell for the leader, ping the Washington node Inband interface.

    Seattle [600: leader] (config) # ping 169.254.145.136
    PING 169.254.145.136 (169.254.145.136) 56(84) bytes of data.
    64 bytes from 169.254.145.136: icmp_seq=1 ttl=64 time=3.44 ms
    64 bytes from 169.254.145.136: icmp_seq=2 ttl=64 time=0.157 ms

Part 5: Enable Cluster on the Joining Node 3

13. Enter `cluster enable` in the command shell on the Boston node.

    Boston (config) # cluster enable
    Boston [600: unknown] (config) #

    > **NOTE:** The transitional state is unknown.

Boston [600: normal] (config) #

> **NOTE:** The normal state indicates that the standby node has completed joining.

14. On the command shell for the leader, enter **show chassis** to ensure all chassis in the "up" state.

Seattle [600: leader] (config) # show chassis
Box# Hostname Config Oper Status HW Type Product# Serial# HW Rev SW Rev
---
7 * Seattle yes up HC3-Chassis 132-0098 80016 A0 3.2.00
8 Washington yes up HC3-Chassis 132-0098 12340 AA 3.2.00
21 Boston yes up HC1-Chassis 132-00A2 40263 A1 3.2.00
Seattle [600: leader] (config) #

15. On the command shell for the leader, enter **show card** to display all line cards in the Inband cluster. Make sure all the line cards are listed.

> **NOTE:** The **show card** command displays all three nodes Inband cluster formation.

Configuration Steps for Node 4: San Francisco

1. Configure node 4, Sanfrancisco as normal in the Inband cluster.
2. Open an SSH or terminal session to the Sanfrancisco node.

Part 1: Using the Jump-Start Wizard to Configure Node 1

3. In the command shell for the Sanfrancisco node, enter the following commands to start the jump-start wizard:
   - enable
   - configure terminal
   - configuration jump-start

gigamon-0d000f > enable
gigamon-0d000f # configure terminal
gigamon-0d000f (config) # configuration jump-start
GigaVUE-OS configuration wizard

4. Enter configuration information for Node 4.

Gigamon GigaVUE-OS
GigaVUE-OS configuration wizard
Do you want to use the wizard for initial configuration? yes
Step 1: Hostname? [gigamon-0d000f] Sanfrancisco
Step 2: Management interface? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no]
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.22/24

Step 6: Default gateway? 10.150.52.1
Step 7: Primary DNS server? 192.168.2.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?

> **NOTE:** In Step 16, accept the default of No so that the cluster is not enabled.

Step 16: Cluster enable? [no]
Step 17: Box-id for the chassis? [1] 22

> **NOTE:** To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

Choice:
Configuration changes saved.
To return to the wizard from the CLI, enter the "configuration jump-start" command from configure mode. Launching CLI...
System in classic mode
Sanfrancisco > enable
Sanfrancisco # configure terminal
Sanfrancisco (config) #

Part 2: Configure the Inband Cluster on the Remote Target Node 4

Node 4, Sanfrancisco, does not have a default cluster interface on GigaVUE-HB1, therefore you do not need to disable zeroconf feature like you would with the other nodes.

Sanfrancisco (config) # cluster interface inband
Sanfrancisco (config) # cluster id 600
Sanfrancisco (config) # cluster name 600
Sanfrancisco (config) # cluster leader address vip 10.150.52.233 /24
Sanfrancisco (config) # interface inband zeroconf
Sanfrancisco (config) #

5. Enter the following command to perform a confirmation check.

Sanfrancisco (config) # show interfaces inband
Interface inband status:
Comment:
Admin up: yes

Link up: yes
DHCP running: no
IP address: 169.254.179.192
Netmask: 255.255.0.0
.
.
.
Sanfrancisco (config) #
Sanfrancisco (config) # show cluster configured
Global cluster config:
Cluster enabled: no
Cluster ID: 600
Cluster name: 600
Cluster control interface: inband

> **NOTE:** The cluster control interface indicates that the cluster is Inband.

Cluster port: 60102
Cluster expected nodes: 1
Cluster startup time: 180
Cluster shared secret: 1234567890123456
Cluster leader preference: 40
Cluster leader auto-discovery enabled: yes
Cluster leader manual port: 60102
Cluster leader virtual IP address: 10.150.52.233/24
Cluster leader management interface: eth0
Sanfrancisco (config) #

Part 3: Configure Relevant Stack Ports and Offline Node 4 Configuration Information

6. Configure the stack ports in the cluster.

   Seattle [600: leader] (config) # card slot 8/5
   Seattle [600: leader] (config) # port 8/5/x1 type stack
   Seattle [600: leader] (config) # port 8/5/x1 params admin enable

7. Configure the offline stack port configuration for Node 4.

   Seattle [600: leader] (config) # chassis box-id 22 serial-num B0020 type hb1
   Seattle [600: leader] (config) # card slot 22/1 product-code 132-00AF
   Seattle [600: leader] (config) # port 22/1/x3 type stack
   ! Box '22' is down, unable to validate SFP type for stack port.

   > **NOTE:** The `Box number is down, unable to validate SFP type for stack port` message is expected behavior.

   Seattle [600: leader] (config) # port 22/1/x3 params admin enable
   Seattle [600: leader] (config) #

8. Enter **`show running-config`** to perform a confirmation check.

Part 4: Configure the Stack Port for the Joining Node 4

9. On the command shell for Node 4, enter the configuration information.

   Sanfrancisco (config) # card slot 22/1
   Sanfrancisco (config) # port 22/1/x3 type stack
   Sanfrancisco (config) # port 22/1/x3 params admin enable
   Sanfrancisco (config) #

   5Enter `show port params` to perform a confirmation check.

   Sanfrancisco (config) # show port params port 22/1/x3
   Parameter 22/1/x3
   ====================== ===============
   Name Alias:
   Type: stack

   > **NOTE:** The stack value indicates the Node 4 port state.

   Admin: enabled
   Link status: up

   > **NOTE:** The Link Status value indicates that the port is “up”.

   Auto Negotiate: off
   Duplex: full
   Speed (Mbps): 10000
   MTU: 9400
   Force Link Up: off
   Port Relay: N/A
   ...

10. On the command shell for the leader, ping the Washington node Inband interface.

    Seattle [600: leader] (config) # ping 169.254.179.192
    PING 169.254.179.192 (169.254.179.192) 56(84) bytes of data.
    64 bytes from 169.254.179.192: icmp_seq=1 ttl=64 time=1.81 ms
    64 bytes from 169.254.179.192: icmp_seq=2 ttl=64 time=0.155 ms
    64 bytes from 169.254.179.192: icmp_seq=3 ttl=64 time=0.136 ms

Part 5: Enable the Cluster on the Joining Node 4

11. Enter `cluster enable` in the command shell of Node 4.

    Sanfrancisco (config) # cluster enable
    Sanfrancisco [600: unknown] (config) #

    > **NOTE:** The transitional state is unknown.

    Sanfrancisco [600: normal] (config) #

> **NOTE:** The “normal” value indicates that the standby node is complete in joining.

12. Enter **show chassis** to display all the chassis in the cluster is in the up state.

```
Seattle [600: leader] (config) # show chassis
Box# Hostname Config Oper Status HW Type Product# Serial# HW Rev SW Rev
--------------------------------------------------------------------------------
7 * Seattle yes up HC3-Chassis 132-0098 80016 A0 3.2.00
8 Washington yes up HC3-Chassis 132-0098 12340 AA 3.2.00
21 Boston yes up HC1P-Chassis 132-00A2 40263 A1 3.2.00
22 Sanfrancisco yes up HC1-Chassis 132-00B1 B0020 3.6 3.2.00
Seattle [600: leader] (config) #
```

13. On the command shell of the leader, enter **show card** to display all line cards in the Inband cluster.

```
Seattle [600: leader] (config) # show card
Seattle [600: leader] (config) #
Seattle [600: leader] (config) # write memory
```

> **NOTE:** The **write memory** command commits the information to the leader database.

Node 4, San Francisco, is completed for Inband cluster formation.

Enable Cluster Management for GigaVUE TA Series Nodes

To enable clustering, GigaVUE TA Series nodes require an Advanced Features License. This license can be obtained by contacting Gigamon Sales team. In order to obtain the license for a Gigamon node, have the node serial number available. All licenses are tied to the serial number and cannot be moved.

For licensing the GigaVUE-OS on a white box, users can access the GigaVUE-OS licensing portal and obtain the license key online. In order to generate the license, the following are required: the serial number of the white box, digital footprint, and Gigamon Installation Key (GIK). The serial number and digital footprint can be obtained by using the following CLI command:

```
(config) # show chassis
```

The GIK is emailed to the customer once the Sales order is placed with Gigamon for the GigaVUE-OS **Advanced Features License**.

How to Apply for Advanced Features License on GigaVUE TA Series Nodes

Once you obtain the license key, use the following CLI command to enable the license:

(config) # license install box-id <box ID> key <license key>

where:

- **Box ID** is the box ID of the GigaVUE TA Series node joining the cluster. Ensure that the box ID of the joining GigaVUE TA Series node does not match any of the existing box IDs in the cluster.
- **License Key** is the license key obtained for the purposes of clustering.

NOTE: This license is different than the license to enable ports on a GigaVUE-OS-TA1.

If using **config jump-start** to enter the wizard mode, on Step xx for the GigaVUE TA Series node, you can also select the option to enable the cluster. To add the license key information, use the same command:

(config) # license install box-id <box ID> key <license key>

Once the license key is enabled, you can join a cluster using Add a Node to an Existing Cluster – Reset to Factory Defaults. Also refer to How to Use Jump-Start Configuration on GigaVUE-OS®TA Series Nodes.

After the Advanced Features Licenses have been installed locally on the GigaVUE-OS-TA1, and the GigaVUE-OS-TA1 joins the cluster, install the GigaVUE-OS-TA1s Advanced Features License again, this time on the leader node.

To enable the license on the leader node as well, follow the same commands on the leader:

(config) # license install box-id <box ID> key <license key>

where:

- **Box ID** is the box ID of the GigaVUE TA Series node joining the cluster.
- **License Key** is the Advanced Features License key for the joining GigaVUE TA Series node.

Installing the Advanced Features License on the leader puts the Advanced Features License in the database.

Inband Cluster Management with GigaVUE TA Series (Including a White Box)

A GigaVUE TA Series node can never be the leader or standby in a cluster configuration. In the example, How to Setup Inband Cluster Management on a New Cluster, if a GigaVUE TA Series node is added to the cluster, it can only take the role of Normal.

Additionally, a license is required to enable cluster formation with a GigaVUE TA Series node. For each GigaVUE TA Series node in a cluster, a specific Advanced Features License is required that is tied to the serial number of that node. In order to add a GigaVUE TA Series node to an Inband cluster, it is important that you have the Advanced Features License installed and enabled on the GigaVUE TA Series node before joining the cluster.

Using the same example as in the section, How to Setup Inband Cluster Management on a New Cluster, we recreated the following example. In it, we add a GigaVUE TA Series node Certified Traffic Aggregation White Box (white box) to establish a stack link from GigaVUE TA Series (8/1/x5..x20) to leader (7/8/x5..x20).

NOTE: The faceplate numbering of a white box is different. Refer to White Box Port and Faceplate Labeling.

In this example, Seattle is the leader. The nodes to be configured in the Inband cluster are as follows:

Node Number	Node Name	Node Type
1	Seattle	GigaVUE-HC3
2	Washington	Certified Traffic Aggregation White Box

NOTE: To configure Inband Cluster Management, you must maintain a command shell for the leader as well as the target nodes, due to the offline configuration that needs to be applied to leader.

Configuration Steps for Leader: Seattle

1Open an SSH or terminal session to the Seattle node.

Part 1: Using the Jump-Start Wizard to Configure Node 1

1. In config, enter `configuration jump-start` to start the jump-start wizard:

```
gigamon-0d0024 > enable
gigamon-0d0024 # configure terminal
gigamon-0d0024 (config) # configuration jump-start
GigaVUE-OS configuration wizard
```

2. Enter the parameter values to configure the leader.

```
Step 1: Hostname? [gigamon-0d0024] Seattle
Step 2: Management interface? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no]
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.150.52.6/24
```

Step 6: Default gateway? 10.150.52.1
Step 7: Primary DNS server? 192.168.2.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?

NOTE: In Step 16, **accept the default of No** so that you do not enable the cluster.

Step 16: Cluster enable? [no]

NOTE: In Step 17, assign the box ID of your chassis.

Step 17: Box-id for the chassis? [1] 7

NOTE: To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

Choice:
Configuration changes saved.
System in classic mode
Seattle (config) #

Part 2: Configuring Inband Cluster on the Leader

3. You need to disable the zeroconf feature on the default cluster interface on eth2 of the control card (HCCv2) in the Seattle node, and make cluster interface Inband with relevant cluster information.

Seattle (config) # no interface eth2 zeroconf
Seattle (config) # cluster interface inband
Seattle (config) # cluster id 600
Seattle (config) # cluster name 600
Seattle (config) # cluster leader address vip 10.150.52.233 /24
Seattle (config) # interface inband zeroconf
Seattle (config) #

4. Enter `show interfaces` to perform a confirmation check.
5. Make sure that no IP address is assigned on eth2 and new IP address is auto assigned for Inband interface.

Seattle (config) # show interfaces

.

.
.
Interface eth2 status:
Comment:
Admin up: yes
Link up: yes
DHCP running: no
IP address:

NOTE: The IP address field on eth2 should be empty.

Netmask:
IPv6 enabled: no
Speed: 1000Mb/s (auto)
.
.
.
Interface inband status:
Comment:
Admin up: yes
Link up: yes
DHCP running: no
IP address: 169.254.51.255

NOTE: The IP address field is automatically assigned.

Netmask: 255.255.0.0
Seattle (config) #

2Enter `show cluster configured` to display the current cluster configuration.

Seattle [600: leader] (config) # show cluster configured
Global cluster config:
Cluster enabled: no
Cluster ID: 600
Cluster name: 600
Cluster control interface: inband

NOTE: The cluster control interface is set to inband.

Cluster port: 60102
Cluster expected nodes: 2
Cluster startup time: 180
Cluster shared secret: 1234567890123456
Cluster leader preference: 60
Cluster leader auto-discovery enabled: yes
Cluster leader manual port: 60102

```
Cluster leader virtual IP address: 10.150.52.233/24
Cluster leader management interface: eth0
Seattle [600: leader] (config) #
```

NOTE: If the jump-start script is used after the Inband cluster is configured, the "inband" option will be available for the cluster interface along with eth0, eth1 and eth2.

Part 3: Configured Leader is Ready for Inband Cluster

6. Enter the card slot and number command. If the designated stack port is located at slot 7, then wait for the card at slot 7 to come "up" to the "oper state."

```
Seattle (config) # card slot 7/8
```

7. Enter **show card** to perform a confirmation check.

```
Seattle [600: leader] (config) # show card
Box ID: 7 (leader)
Slot Config Oper Status HW Type Product Code Serial Num HW Rev
--------------------------------------------------------------------------------
1 no inserted GigaPORT-X12G04 132-0045 1450-0162 B4-a6
2 no inserted GigaPORT-C01 132-00A8 1A80-0114 A2-5
3 no inserted GigaPORT-Q08 132-00AK 1AK0-0022 A0-a4
4 no inserted GigaPORT-X12G04 132-0045 1450-0181 B4-a6
cc1 yes up H-CCv2 132-0089 1890-0036 5.2-df
5 no inserted GigaPORT-X12G04 132-0045 1450-0224 C2-a6
6 no inserted GigaPORT-Q02X32 132-0087 1870-0157 B2-a1
7 no inserted GigaPORT-C01 132-00A8 1A80-0107 A2-5
8 yes up GigaPORT-Q02X32/2q 132-0087 1870-0167 B2-a2
```

Seattle [600: leader] (config) #

8. Assign and enable the stack ports. Make them stack GigaStream.
9. Enable the cluster.

NOTE: The Seattle node is now the leader as indicated in the CLI prompt.

```
Seattle (config) # port 7/8/x5..x20 type stack
Seattle (config) # port 7/8/x5..x20 params admin enable
Seattle (config) # gigastream alias big_bridge_7to8 port 7/8/x5..x20
Seattle (config) # cluster enable
Seattle [600: leader] (config) #
```

Part 4: Apply Offline Remote Node Configuration on the Leader

10. Apply offline remote node configuration on the leader as shown in the CLI command to conclude leader setup.

```
Seattle [600: leader] (config) # chassis box-id 8 serial-num QTFCEA4170003 type ly2r
Seattle [600: leader] (config) # card slot 8/1 product-code 132-0087 mode 48x
```

NOTE: It is imperative to add the mode configured on the GigaVUE TA Series node that is joining the cluster to ensure that the leader has the correct configuration. Refer to Offline Remote Configuration for GigaVUE-OS® TA Series Nodes.

Seattle [600: leader] (config) # port 8/1/x5..x20 type stack
! Box '8' is down, unable to validate SFP type for stack port.

NOTE: The **box number is down, unable to validate SFP type for stack port** message is expected behavior.

.
.
.

Seattle [600: leader] (config) # port 8/1/x5..x20 params admin enable
Seattle [600: leader] (config) # gigastream alias big_bridge_8to7 port 8/1/x5..x20
Seattle [600: leader] (config) # write memory

3Enter `show running-config` to perform a confirmation check.

Seattle [600: leader] (config) # show running-config

Configure Inband Cluster on the Remote Target Node: Certified Traffic Aggregation White Box: Washington

Part 1: Using the jump-start wizard to configure the node:

Do you want to use the wizard for initial configuration? yes
Step 1: Hostname? [gigamon-5508f4] Washington
Step 2: Management Interface <eth0> ? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no] no
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.115.26.114 /21
Step 6: Default gateway? 10.115.24.1
Step 7: Primary DNS server? 10.10.1.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?
No valid advanced features license found!
Step 16: Box-id for the chassis? [1] 8

```
You have entered the following information:
1. Hostname: Washington
2. Management Interface <eth0> : eth0
3. Use DHCP on eth0 interface: no
4. Use zeroconf on eth0 interface: no
5. Primary IPv4 address and masklen: 10.115.26.114/21
6. Default gateway: 10.115.24.1
7. Primary DNS server: 10.10.1.20
8. Domain name: gigamon.com
9. Enable IPv6: yes
10. Enable IPv6 autoconfig (SLAAC) on eth0 interface: no
11. Enable DHCPv6 on eth0 interface: no
12. Enable secure cryptography: no
13. Enable secure passwords: no
14. Minimum password length: 8
16. Admin password: ********
17. Box-id for the chassis: 8
To change an answer, enter the step number to return to.
Otherwise hit <enter> to save changes and exit.
Choice:
Configuration saved to database 'initial'
Configuration changes saved.
```

Part 2: Configure the Inband Cluster on the Remote Target GigaVUE TA Series Node

```
Washington (config) # card slot 8/1
Washington (config) # license install box-id 8 key <Port_License>
License installed successfully on slot 8/1
Washington (config) # license install box-id 8 key <Advanced_Features_License>
License installed successfully on slot 8/1
Washington (config) # cluster interface inband
Washington (config) # cluster id 600
Washington (config) # cluster name 600
Washington (config) # cluster leader address vip 10.115.26.151 /21
Washington (config) # interface inband zeroconf
Washington (config) # card slot 8/1
Washington (config) # port 8/1/x5..x20 type stack
Washington (config) # port 8/1/x5..x20 params admin enable
Washington (config) # gigastream alias big_bridge_8to7 port-list 8/1/x5..x20
Washington (config) # write memory
Configuration saved to database 'initial'
```

Enter show cluster configured to display the cluster configuration settings.

Make sure that the “Inband” value is defined in the cluster control interface.

```
Washington (config) # show cluster configured
Global cluster config:
Cluster enabled: no
Cluster ID: 600
Cluster name: 600
Cluster control interface: inband
```

NOTE: The cluster control interface is set to “inband”.

```
Cluster port: 60102
Cluster expected nodes: 1
Cluster startup time: 180
Cluster shared secret: 1234567890123456
Cluster leader preference: 0
Cluster leader auto-discovery enabled: yes
Cluster leader manual port: 60102
Cluster leader virtual IP address: 10.115.26.151/21
Cluster leader management interface: eth0
```

NOTE: Stack interface should show as up.

```
Washington (config) # show port params port-list 8/1/x5
Parameter 8/1/x5
 ====================== ===============
Name Alias:
Type: stack
Admin: enabled
Link status: up
```

NOTE: The Link Status indicates that the stack port is “up” state.

```
Auto Negotiate: off
Duplex: full
Speed (Mbps): 10000
MTU: 9400
Force Link Up: off
Port Relay: N/A
...
```

Now that you have established the port license and Advanced Features License on the GigaVUE TA Series, move to the leader in the cluster and run the following steps. (The following example shows the steps on a GigaVUE-OS-TA1.)

Part 3: Enable the Cluster on the Joining GigaVUE TA Series Node

```
Washington (config) # cluster enable
Washington [600: unknown] (config) #
```

NOTE: The Washington node is in an unknown transitional state.

```
Washington [600: normal] (config) #
```

Part 4: Finish Configuration of the Cluster on the Leader (Only Required with GigaVUE TA Series Nodes Including White Box)

NOTE: Ensure GigaVUE TA Series node or the white box has joined the cluster and the chassis is up.

```
Seattle [600: leader] # show chassis
```

For white box nodes, the Quanta box information is shown as HW Type. This will vary with the GigaVUE-OS nodes.

Install the Port License of the GigaVUE TA Series Node (Normal) on Leader

```
Seattle [600: leader] (config) # license install box-id 8 key <Port_License>
License installed successfully on slot 8/1
Seattle [600: leader] (config) # show license
---------
Box 1
---------
---------
Box 8
---------
Slot Feature Parameters Expiration Date
----------------------------------------
1 PORT - Never
Chassis-Feature Parameters
--------------------------
CLUSTER -
```

Check the License on Washington Node

```
System in classic mode
Washington [600: normal] > en
Washington [600: normal] # show license
---------
Box 8
---------
Slot Feature Parameters Expiration Date
----------------------------------------
1 PORT - Never
Chassis-Feature Parameters
--------------------------
CLUSTER -
```

Set up Inband Cluster Management with GigaVUE-OS TA-100 or GigaVUE-HC3

In a multi-node Inband Cluster Management, GigaVUE-OS TA-100 or GigaVUE-HC3 can be added as a node located at the middle of a star, daisy-chain, or tree topology.

In this example, Seattle is the leader. The nodes to be configured in the Inband cluster are as follows:

Node Number	Node Name	Node Type
1	Seattle (leader)	GigaVUE-HC3
2	Washington (normal)	GigaVUE-OS-TA100
3	San Francisco	GigaVUE-HC1-Plus

NOTE: To configure Inband Cluster Management, you must maintain a command shell for the leader as well as the target nodes, due to the offline configuration that needs to be applied to leader.

Configuration Steps for Leader: GigaVUE-HC3: Seattle

1. Open an SSH or terminal session to the Seattle node.
2. In Config mode, enter configuration jump-start to start the jump-start wizard:

```
gigamon-0d0024 > enable
gigamon-0d0024 # configure terminal
gigamon-0d0024 # configuration jump-start
GigaVUE-OS configuration wizard
```

3. Enter the parameter values to configure the leader.

```
Step 1: Hostname? [gigamon-5508f4] Seattle
Step 2: Management Interface <eth0 eth2 eth3> ? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no] no
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.115.26.114 /21
Step 6: Default gateway? 10.115.24.1
Step 7: Primary DNS server? 10.10.1.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
```

```
Step 15: Confirm admin password?
```

NOTE: In Step 16, accept the default of No so that you do not enable the cluster.

```
Step 16: Cluster enable? [no]
```

NOTE: In Step 17, assign the box ID of your chasis.

```
Step 17: Box-id for the chassis? [1] 10
```

NOTE: To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

```
Choice:
Configuration changes saved.
System in classic mode
Seattle (config) #
```

4. Disable the Zeroconf feature on the default cluster interface on eth2 of the control card (HCCv2) in the Seattle node, and make cluster interface Inband with relevant cluster information.

```
System in classic mode
Seattle (config) # no interface eth2 zeroconf
Seattle (config) # cluster interface inband
Seattle (config) # cluster leader preference 100
Seattle (config) # cluster id 600
Seattle (config) # cluster name 600
Seattle (config) # cluster leader address vip 10.115.26.151 /21
Seattle (config) # interface inband zeroconf
```

5. Enable the card. To enable the card enter the card slot and number command.

```
Seattle (config) # card slot 10/3
```

6. Assign and enable the stack ports.

```
Seattle (config) # port 10/3/c2 type stack
Seattle (config) # port 10/3/c2 params admin enable
```

7. Enable the cluster.

```
Seattle (config) # cluster enable
```

Apply Offline Remote Node Configuration on the Leader

8. Apply offline remote node configuration on the leader as shown in the CLI command.

```
Seattle (config) # chassis box-d 11 serial num F002E type itac
Seattle (config) # card slot 11/1 product-code 132-00CG
Seattle (config) # port 10/3/c2, 11/1/c23 type stack
Seattle (config) # port 10/3/c2, 11/1/c23 params admin enable
Seattle (config) # stack-link alias hc3-10-to-ta100-11 between port 10/7/c2 and
11/1/c23
Seattle (config) # chassis box-d 3 serial num 40201 type hc3-ccv2
Seattle (config) # card slot 3/3 product-code 132-00BY
Seattle (config) # port 3/3/c6, 11/1/c17 type stack
Seattle (config) # port 3/3/c6, 11/1/c17 params admin enable
Seattle (config) # stack-link alias ta100-11-to-hc3-3 between port 11/1/c17 and
3/3/c6
```

```
Seattle (config) # write memory
```

Configuration Steps for Standby Node: San Francisco

1. Open an SSH or terminal session to the San Francisco node.

Part 1: Configure Inband Cluster on the Standby Node: GigaVUE-HC1-Plus: San Francisco

2. In config, enter `configuration jump-start` to start the jump-start wizard:

```
gigamon-0d0024 > enable
gigamon-0d0024 # configure terminal
gigamon-0d0024 (config) # configuration jump-start
GigaVUE-OS configuration wizard
```

3. Enter the parameter values to configure the standby node.

```
Step 1: Hostname? [gigamon-5508f4] Sanfrancisco
Step 2: Management Interface <eth0 eth2 eth3> ? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no] no
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.115.26.114 /21
Step 6: Default gateway? 10.115.24.1
Step 7: Primary DNS server? 10.10.1.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?
No valid advanced features license found!
```

NOTE: In Step 16, **accept the default of No** so that you do not enable the cluster.

```
Step 16: Cluster enable? [no]
```

NOTE: In Step 17, assign the box ID of your chassis.

```
Step 17: Box-id for the chassis? [1] 3
```

NOTE: To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

```
Choice:
Configuration changes saved.
System in classic mode
Sanfrancisco (config) #
```

Part 3: Configuring Inband Cluster on the Standby Node

4. You need to disable the zeroconf feature on the default cluster interface on eth2 of the control card (HCCv2) in the Sanfrancisco node, and make cluster interface Inband with relevant cluster information.

```
Sanfrancisco (config) # no interface eth2 zeroconf inband
Sanfrancisco (config) # cluster interface inband
Sanfrancisco (config) # cluster leader preference 60
Sanfrancisco (config) # cluster id 600
Sanfrancisco (config) # cluster name 600
Sanfrancisco (config) # cluster leader address vip 10.115.26.151 /21
Sanfrancisco (config) # interface inband zeroconf
```

5. Enable card. To enable the card, enter the card slot and number command.

```
Sanfrancisco (config) # card slot 3/3
```

6. Assign and enable the stack ports.

```
Sanfrancisco (config) # port 3/3/c6 type stack
Sanfrancisco (config) # port 3/3/c6 params admin enable
```

7. Enable the cluster.

```
Sanfrancisco (config) # cluster enable
```

8. Apply offline remote node configuration on the standby node as shown in the CLI command.

```
Sanfrancisco (config) # chassis box-d 11 serial num F002E type itac
Sanfrancisco (config) # card slot 11/1 product-code 132-00CG
Sanfrancisco (config) # port 11/1/c17 type stack
Sanfrancisco (config) # port 11/1/c17 params admin enable
Sanfrancisco (config) # stack-link alias hc3-3-to-ta100-11 between port 3/3/c6
and 11/1/c17
Sanfrancisco (config) # write memory
```

Part 2: Configure the Inband Cluster on the GigaVUE TA Series Node: GigaVUE-TA100: Washington

1. Open an SSH or terminal session to the Washington node.
2. In config, enter `configuration jump-start` to start the jump-start wizard:

```
gigamon-5508f4 > enable
gigamon-5508f4 # configure terminal
gigamon-5508f4 (config) # configuration jump-start
GigaVUE-OS configuration wizard
```

3. Enter the parameter values to configure the GigaVUE-OS TA100 Series node.

```
Step 1: Hostname? [gigamon-5508f4] Washington
Step 2: Management Interface <eth0> ? [eth0]
Step 3: Use DHCP on eth0 interface? no
Step 4: Use zeroconf on eth0 interface? [no] no
Step 5: Primary IPv4 address and masklen? [0.0.0.0/0] 10.115.26.114 /21
Step 6: Default gateway? 10.115.24.1
Step 7: Primary DNS server? 10.10.1.20
Step 8: Domain name? gigamon.com
Step 9: Enable IPv6? [yes]
```

```
Step 10: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no]
Step 11: Enable DHCPv6 on eth0 interface? [no]
Step 12: Enable secure cryptography? [no]
Step 13: Enable secure passwords? [no]
Step 14: Minimum password length? [8]
Step 15: Admin password?
Please enter a password. Password is a must.
Step 15: Admin password?
Step 15: Confirm admin password?
No valid advanced features license found!
```

NOTE: In Step 16, **accept the default of No** so that you do not enable the cluster.

```
Step 16: Cluster enable? [no]
```

NOTE: In Step 17, assign the box ID of your chassis.

```
Step 17: Box-id for the chassis? [1] 11
```

NOTE: To change the answers in the jump-start wizard, enter the step number that you want to change. Click Enter to save changes and exit.

```
Choice:
Configuration changes saved.
System in classic mode
Washington (config) #
```

4. Apply cluster and inband interface settings on the TA00 series node.

```
Washington (config) # cluster interface inband
Washington (config) # cluster id 600
Washington (config) # cluster name 600
Washington (config) # cluster leader address vip 10.115.26.151 /21
Washington (config) # interface inband zeroconf
```

5. Enable the card. To enable the card, enter the card slot and number command.

```
Washington (config) # card slot 11/1
```

6. Assign and enable the stack ports.

```
Washington (config) # port 11/1/c17, 11/1/c23 type stack
Washington (config) # port 11/1/c17, 11/1/c23 params admin enable
```

7. Enable the cluster.

```
Washington (config) # cluster enable
```

8. Apply offline remote node configuration on the GigaVUE-OS TA100 series node as shown in the CLI command.

```
Washington (config) # chassis box-d 10 serial num 40201 type hc3-ccv2
Washington (config) # card slot 10/3 product-code 132-00BY
Washington (config) # port 10/3/c2 type stack
Washington (config) # port 10/3/c2 params admin enable
Washington (config) # stack-link alias hc3-10-to-ta100-11 between port 10/3/c2
and 11/1/c17
Washington (config) # chassis box-d 3 serial num 40201 type hc3-ccv2
Washington (config) # card slot 3/3 product-code 132-00BY
Washington (config) # port 3/3/c6 type stack
Washington (config) # port 3/3/c6 params admin enable
```

```
Washington (config) # stack-link alias ta100-11-to-hc3-3 between port 11/1/c17
and 3/3/c6
Washington (config) # write memory
```

Part 2: Configuring Inband Cluster on the GigaVUE TA Series Node

9. You need to disable the zeroconf feature on the default cluster interface on eth2 of the control card (HCCv2) in the Seattle node, and make cluster interface Inband with relevant cluster information.

```
Washington (config) # no cluster interface inband
Washington (config) # cluster leader preference 60
Washington (config) # cluster id 600
Washington (config) # cluster name 600
Washington (config) # cluster leader address vip 10.115.26.151 /21
Washington (config) # interface inband zeroconf
System in classic mode
Washington (config) # card slot 11/1
Washington (config) # port 11/1/c17, 11/1/c23 type stack
Washington (config) # port 11/1/c17, 11/1/c23 params admin enable
```

10. Apply offline remote node configuration on the GigaVUE-OS TA100 series node as shown in the CLI command.

```
Washington (config) # chassis box-d 10 serial num 40201 type hc3-ccv2
Washington (config) # card slot 10/3 product-code 132-00BY
Washington (config) # port 10/3/c2 type stack
Washington (config) # port 10/3/c2 params admin enable
Washington (config) # stack-link alias hc3-10-to-ta100-11 between port 10/3/c2
and 11/1/c17
Washington (config) # chassis box-d 3 serial num 40201 type hc3-ccv2
Washington (config) # card slot 3/3 product-code 132-00BY
Washington (config) # port 3/3/c6 type stack
Washington (config) # port 3/3/c6 params admin enable
Washington (config) # stack-link alias ta100-11-to-hc3-3 between port 11/1/c17
and 3/3/c6
Washington (config) # write memory
```

How to Switch from Inband Cluster Management to Out-of-Band

The following example shows the CLI commands to switch from an Inband to an Out-of-Band cluster for aGigaVUE HC Series node.

CLI command syntax:

```
cluster interface eth0 | eth2
interface eth2 zeroconf
no interface inband zeroconf
```

1. Open a command shell in the leader node, Seattle (GigaVUE-HC3) and run the following CLI commands:

```
Seattle [600: leader] (config) # cluster interface eth0
Seattle [600: *unknown*] (config) # no interface inband zeroconf
Seattle [600: *unknown*] (config) # interface eth2 zeroconf
Seattle [600: *unknown*] (config) #
```

2. Open a command shell in the Boston node (HC3) and run the following CLI commands:

```
Boston [600: leader] (config) # cluster interface eth0
Boston [600: *unknown*] (config) # interface eth2 zeroconf
Boston [600: *unknown*] (config) # no interface inband zeroconf
Boston [600: *unknown*] (config) #
```

The following example shows the CLI commands to switch from an Inband to an Out-of-Band cluster for a GigaVUE HC Series nodes.

CLI command syntax:

```
cluster interface eth1
interface eth1 zeroconf
no interface inband zeroconf
```

3. Open a command shell in the Washington node (GigaVUE-HC3) and run the following CLI commands:

```
Washington [600: leader] (config) # cluster interface eth1
band zeroconf
Washington [600: *unknown*] (config) # interface eth1 zeroconf
Washington [600: *unknown*] (config) # no interface inband zeroconf
Washington [600: *unknown*] (config) #
```

The following example shows the CLI commands to switch from an Inband to an Out-of-Band cluster for a GigaVUE-HC1 node.

CLI command syntax:

```
cluster interface eth0
no interface inband zeroconf
```

4. Open a command shell in the Sanfrancisco node (GigaVUE-HC1) and run the following CLI commands:

```
Sanfrancisco [600: leader] (config) # cluster interface eth0
Sanfrancisco [600: *unknown*] (config) # no interface inband zeroconf
Sanfrancisco [600: *unknown*] (config) #
```

5. Enter `show chassis` to display the chassis oper status.

```
Seattle [600: leader] (config) # show chassis
```

NOTE: The eth0 value replaces the Inband value to confirm the switch.

How to Switch from Out-of-Band to Inband Cluster Management

The following example shows the CLI commands to switch from an Out-of-Band to an Inband cluster for a GigaVUE HC Series node with CCv2 control card(s).

CLI command syntax:

```
cluster interface inband
no interface eth2 zeroconf
interface inband zeroconf | interface inband <ip address / ip mask>
```

1. Open a command shell in the leader node, Seattle (GigaVUE-HC3) and run the following CLI commands:

```
Seattle [600: leader] (config) # cluster interface inband
Seattle [600: *unknown*] (config) # interface inband zeroconf
Seattle [600: *unknown*] (config) # no interface eth2 zeroconf
Seattle [600: *unknown*] (config) #
```

2. Open a command shell in the Boston node (HC3) and run the following CLI commands:

```
Boston [600: standby] (config) # cluster interface inband
Boston [600: *unknown*] (config) # interface inband zeroconf
Boston [600: *unknown*] (config) # no interface eth2 zeroconf
Boston [600: *unknown*] (config) #
```

The following example shows the CLI commands to switch from an Out-of-Band to Inband cluster for a GigaVUE HC Seriesnode with CCv2 control card(s).

CLI command syntax:

```
cluster interface inband
no interface eth1 zeroconf
interface inband zeroconf | interface inband <ip address / ip mask>
```

3. Open a command shell in the Washington node (GigaVUE-HC3 CCv2) and run the following CLI commands:

```
Washington [600: standby] (config) # cluster interface inband
Washington [600: *unknown*] (config) # interface inband zeroconf
Washington [600: *unknown*] (config) # no interface eth1 zeroconf
Washington [600: *unknown*] (config) #
```

The following example shows the CLI commands to switch from an Out-of-Band to an Inband cluster for a GigaVUE-HC1 node.

CLI command syntax:

```
cluster interface eth0
no interface inband zeroconf
```

4. Open a command shell in the Sanfrancisco node (GigaVUE-HC1) and run the following CLI commands:

```
Sanfrancisco [600: leader] (config) # cluster interface inband
Sanfrancisco [600: *unknown*] (config) # interface inband zeroconf
Sanfrancisco [600: *unknown*] (config) #
```

5. Enter `show chassis` to display the chassis oper status.

NOTE: The Inband value in the output confirms the switch.

How to Remove a Node from an IPv4 Cluster and Add it to an IPv6 Cluster and Vice-versa

You can remove a node from an existing IPv4 cluster and add it to another IPv6 cluster. Refer to the following examples.

Physical Devices Out of Band Clusters

In physical devices OOB cluster, the nodes are auto-discovered. In the following example:

- Seattle 600 is an IPv4 cluster
- Seattle 100 is an IPv6 cluster
- Node1chassis-member is a node that is present in cluster Seattle 600 that must be moved to Seattle 100

Run the following CLI commands:

Step	Command
To remove Node1chassis-member from the cluster Seattle 600	`Seattle [600: Node1chassis-member] (config) # no cluster enable`
To configure the IP protocol of Node1chassis-member as IPv6	`Node1chassis-member (config) # cluster interface eth0 ipv6`

To add Node1chassis-member to cluster Seattle 100	`Node1chassis-member (config) # cluster id 100`
	`Node1chassis-member (config) # cluster name 100`
	`Seattle [100: Node1chassis-member] (config) # cluster enable`

The node is removed from cluster 600 which is an IPv4 cluster and added to the cluster 100 which is an IPv6 cluster.

Layer 3 Out of Band Clusters

In Layer 3 Out Of Band cluster, the nodes must be manually auto-discovered. In the following example:

- Seattle 600 is an IPv4 cluster
- Seattle 100 is an IPv6 cluster
- Node1chassis-member is a node that is present in cluster Seattle 600 that must be moved to Seattle 100

Run the following CLI commands:

Step	Command
To remove Node1chassis-member from the cluster Seattle 600	`Seattle [600: Node1chassis-member] (config) # no cluster enable`
To remove the primary and secondary IP address	`Node1chassis-member (config) # no cluster leader primary ip` `Node1chassis-member (config) # no cluster leader secondary ip`
To configure the IP protocol of Node1chassis-member as IPv6	`Node1chassis-member (config) # cluster interface eth0 ipv6`

To add Node1chassis-member to cluster Seattle 100	`Node1chassis-member (config) # cluster id 100`
	`Node1chassis-member (config) # cluster name 100`
	`Seattle [100: Node1chassis-member] (config) # cluster enable`

The node is removed from cluster 600 which is an IPv4 cluster and added to the cluster 100 which is an IPv6 cluste

Troubleshooting

The following troubleshooting scenarios describe configuration issues that may occur while setting up Inband Cluster Management.

1. To test the Inband communication between two Inband clustering nodes, ping the Inband interface of the target node from the leader node. The following example illustrates this scenario:

```
Seattle [600: leader] (config) # ping 169.254.179.192
PING 169.254.179.192 (169.254.179.192) 56(84) bytes of data.
64 bytes from 169.254.179.192: icmp_seq=1 ttl=64 time=1.81 ms
64 bytes from 169.254.179.192: icmp_seq=2 ttl=64 time=0.155 ms
64 bytes from 169.254.179.192: icmp_seq=3 ttl=64 time=0.136 ms
```

2. To test the database of the leader and the target node before a join, use the `show running-config` to validate database context of both nodes. The leader database must include configuration of the joining node. Otherwise, the joining node will fail to join with “unknown” cluster role.
3. If two leader nodes occur during a new node joining, the cluster will result in split-brain symptom. The joining node with the empty database will become the new leader. The original cluster database can be wiped out by the empty database. It is recommended that you rebuild the cluster. Use the `reset factory only-traffic` command to clean up the database while preserving the management IP address. Next use the `no cluster enable` command to put all nodes in standalone mode. Finally, begin to rebuild the cluster as discussed in the How to Setup Inband Cluster Management on a New Cluster.

Handling System Failure in a Cluster Environment

Every node in a cluster environment has its own copy of the current database specifying all aspects of packet distribution configuration. If a leader ever does go down, the standby node automatically takes possession of the leader VIP address so the cluster can remain operative.

Cluster Commands

The **cluster** and **stack-link** commands are used to establish and manage clusters. Refer to the following commands :

- cluster
- stack-link

In addition, there are a wide variety of **show** commands for clusters, including:

Show Command	Description
show cluster box-id	Displays cluster configuration for a specific box ID. Box IDs identify a specific chassis in the cluster by associating a static identifier (the box-id) with the chassis serial number.
show cluster configured	Displays global cluster configuration state and information.
show cluster global	Displays global cluster run state.
show cluster history	Displays cluster history log. For more information, refer to Cluster Diagnostics.
show cluster local	Displays local cluster run state.
show cluster leader	Displays run state information about leader.
show cluster node	Displays cluster configuration for a specific node ID. In

Show Command	Description
	contrast to the box ID, node IDs are assigned dynamically by the cluster leader and change when a node leaves and rejoins the cluster.
show cluster standby	Displays run state information about standby node.

Cluster-Wide and Local Commands

When working with a cluster, most configuration settings made on one node are synchronized to all other nodes in the cluster, resulting in a seamless, unified Deep Observability Pipeline. On the other hand, some settings are kept locally and only apply to the local node. The following table summarizes the commands that only apply to a local node and those that are pushed to all nodes in the cluster.

NOTE: A good rule of thumb is that if a command takes a box ID as part of its arguments, it can be configured for an individual node from the cluster's leader/VIP address. If a command does not take a box ID as part of its arguments, the corresponding setting must be configured from the individual clustered nodes.

Global Cluster Commands
The following commands are synchronized with all nodes in the cluster, regardless of whether they are made from the VIP or an individual node in the cluster.
map
map-passall
port-pair
tool-mirror
gigastream
hosts
time (clock)
snmp hosts and notifications You configure SNMP hosts and notification events from the leader. The settings are pushed to each node. However, when a clustered node sends an SNMP notification, it is sent from its own Mgmt port, not from the leader/VIP address. In addition, you browse each individual clustered node's MIB separately, not over the VIP/leader.
cli session auto-logout

Global Cluster Commands
ssh server enable/disable
web
ssh host-key **Note**: The leader node automatically pushes its SSH hostkeys to all other nodes in the cluster, ensuring that they are the same on all nodes. If you connected through SSH to perform the initial configuration of a normal node, it will have different keys after this synchronization, likely resulting in a warning from your SSH client when you log back into that individual node later on. This is normal behavior.
syslog settings Similar to SNMP notifications, syslog messages are sent from each individual reporting node's Mgmt port and not from the leader/VIP address.
aaa settings (RADIUS, TACACS+, and LDAP, including AAA)
ntp settings **Note**: The configuration is updated from the leader.

Local Commands
The following commands must be made on a local node – they are not synchronized from the leader.
hostname
ip settings for Mgmt port **Note**: Although you configure a clustered node's IP settings for the Mgmt port over its local console port, the settings once made are stored in the global configuration database along with node's box ID.
ptp settings

Configure External Authentication in a Cluster

Use external authentication with a AAA server for GigaVUE-OS nodes operating in a cluster. AAA settings (RADIUS, TACACS+, and LDAP) are configured on the leader node, and are then synchronized with the other nodes in the cluster.

The nodes in the cluster must be reachable on the network. Specify the virtual IP (VIP) address on the leader (not the local Mgmt IP address).

To use external authentication with a cluster, perform the following steps:

1. On the leader, configure AAA settings:

a. Enable **radius**, **tacacs+**, or **ldap** with the **aaa authentication** command.Refer to the *"Configure AAA Authentication Options"* section in the *GigaVUE Administration Guide* for detailed information.

b. Add the RADIUS, TACACS+, or LDAP server to the GigaVUE-OS node's list using the corresponding **radius**, **tacacs-server**, or **ldap** command. Refer to the *"Add AAA Servers to the Node's List"* section in the *GigaVUE Administration Guide* for more details.

2. Set up users within the external authentication server itself.
3. Add the IP addresses of the Mgmt ports for each node in the cluster to the external authentication server. For example, in Cisco ACS, you add these IP addresses using the **AAA Client IP Address** field in the **Network Configuration** tab.
4. Set up roles/permissions for users in the external authentication server. Refer to the *"Grant Roles with External Authentication Servers"* in the *GigaVUE Administration Guide* for more details.

Once you have configured these settings, you will be able to log in to the cluster over the VIP address using AAA.

Cluster Diagnostics

To diagnose cluster-related issues, you can display the cluster membership history. The **show cluster history** command displays a history of the most recent 200 cluster-related events for a node. The cluster membership events include joins, leaves, membership updates, and initial configuration synchronizing.

The cluster membership history includes the following information (also found in the system log file):

Table 22: Cluster Membership History Information

Name	Format
[Index]	The entry index. The [0] entry is always displayed and provides information about when the log was started. The circular log maintains up to another 199 entries. For example: [140]
Timestamp	The date and time of the event, including the millisecond. The format is YYYY/MM/DD for the date, HH:MM:SS for the time, and .xxx for the millisecond. For example: 2015/02/06 10:47:19.918
Log Event	The short description of the event.

Name	Format
	For example: cfg sync
Cluster Node ID	The cluster node ID. Zero (0) is valid until a node is assigned a node ID. For example: 3
Current role	The name of current role. The valid roles are: unknown, leader, standby, and normal. In a stable cluster, there will be one leader, one standby, and the remaining nodes will be normal. For example: leader
Action	The detailed description of the event. For example: Cfg sync resp msg received (src nodeid 6, dst nodeid 3)

Use the following CLI command on a node in the cluster to display the cluster membership history for that node:

(config) # show cluster history

Use the following CLI command on the leader to display the cluster membership history for a specific node in the cluster:

(config) # show cluster history box-id 2

If you use the following command when you are not on the leader, an error message is displayed:

(config) # show cluster history box-id 2

```
Not leader - can only display cluster log for local box (1).
```

Configure Multi-Path Leaf and Spine

The leaf and spine architecture is a two-layer architecture used for network aggregation. There are two kinds of nodes in this architecture, as follows:

- leaf nodes, which are edge nodes and can also have TAPs or tools attached to them
- spine nodes, which are the nodes to which the leaf nodes attach

Related Topics

- Refer to the "*Multi-path Leaf and Spine*" chapter in the *GigaVUE Fabric Management Guide* for information about creating and managing a map.
- Refer to stack-link in the reference section for details on the syntax of the stack link CLI command.

- Refer to spine-link in the reference section for details on the syntax of the spine link CLI command.

CLI Configuration Example

The figureFigure 27Leaf Spine Topology shows the topology for this configuration example.

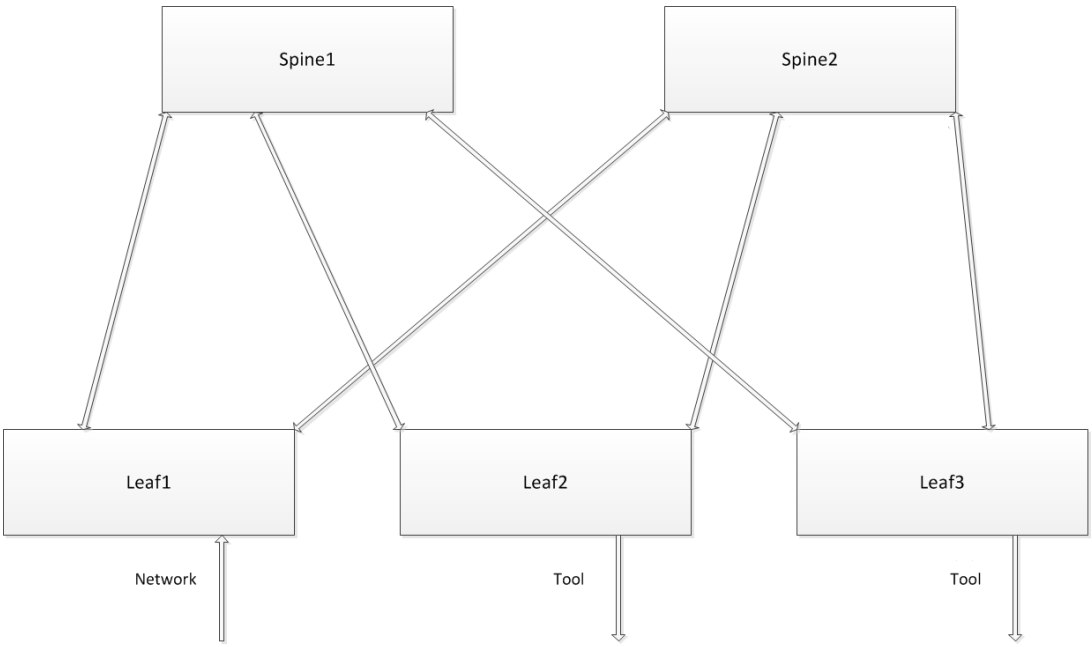


Figure 27 *Leaf Spine Topology*

It is assumed that the out-of-band cluster has already been configured prior to this configuration example.

One of the nodes in the cluster is the leader. The configuration steps are done from the leader node.

TheFigure 27Leaf Spine Topology, incoming traffic arrives on Leaf1. Outgoing traffic is sent to tools on Leaf2 and Leaf3. On the right side, the GigaStream between Leaf3 and Spine2 have multiple ports, forming a larger trunk.

The abbreviations of the aliases used in this configuration example are as follows:

- l1, l2, and l3 refer to leaf1, leaf2, and leaf3 nodes
- s1 and s2 refer to spine1 and spine2 nodes
- gs refers to GigaStream
- sl refers to stack link

The following are the stack GigaStream. There are 12 in all, as follows:

- leaf1, spine1 GigaStream (l1s1gs)
- leaf1, spine2 GigaStream (l1s2gs)

- leaf2, spine1 GigaStream (l2s1gs)
- leaf2, spine2 GigaStream (l2s2gs)
- leaf3, spine1 GigaStream (l3s1gs)
- leaf3, spine2 GigaStream (l3s2gs)
- spine1, leaf1 GigaStream (s1l1gs)
- spine1, leaf2 GigaStream (s1l2gs)
- spine1, leaf3 GigaStream (s1l3gs)
- spine2, leaf1 GigaStream (s2l1gs)
- spine2, leaf2 GigaStream (s2l2gs)
- spine2, leaf3 GigaStream (s2l3gs)

The following are the spine links. There are 3 in all, as follows:

- leaf1spine (l1spine), consisting of l1s1gs and l1s2gs
- leaf2spine (l2spine), consisting of l2s1gs and l2s2gs
- leaf3spine (l3spine), consisting of l3s1gs and l3s2gs

The following are the stack links. There are 6 in all, as follows:

- leaf1, spine1 stack link (l1s1sl), between l1s1gs and s1l1gs
- leaf2, spine1 stack link (l2s1sl), between l2s1gs and s1l2gs
- leaf3, spine1 stack link (l3s1sl), between l3s1gs and s1l3gs
- leaf1, spine2 stack link (l1s2sl), between l1s2gs and s2l1gs
- leaf2, spine2 stack link (l2s2sl), between l2s2gs and s2l2gs
- leaf3, spine2 stack link (l3s2sl), between l3s2gs and s2l3gs

CLI Configuration Commands

Use the following CLI commands to configure the multi-path leaf and spine architecture of the nodes in a cluster environment, in the following order:

- Define stack GigaStream.
- Define spine links.
- Define stack links.

Step	Description	Command
1.	Configure five stack ports.	**(config) # port <port-list> type stack** **(config) # port <port-list> param admin enable**
2.	Configure network and tool ports.	**(config) # port <port-id> type network** **(config) # port <port-id> type tool**

Step	Description	Command
3.	Configure stack GigaStream. In this example, 12 GigaStreams have been configured.	**(config) # gigastream alias <gigastream-alias> port-list <port-list>**
4.	Configure spine links. There are 3 spine links required for this configuration example. The spine links are located at the leaf nodes.	**(config) # spine-link <spine-link alias> port-list <port-list>**
5.	Configure stack links. There are 6 stack links required for this configuration example.	**(config) # stack-link <stack-link alias> between gigastreams <gigastream alias> and <gigastream alias>**
6.	Configure a map, from a network port to tool ports.	**(config) # map alias <map alias>** **(config map alias map1) # from <port-list>** **(config map alias map1) # to <port-list>** **(config map alias map1) # rule add pass protocol tcp** **(config map alias map1) # exit** **(config) #**

NOTE: Configure ports of same speed in the Stack GigaStream.

Configure GigaVUE HC Series Security Options

The GigaVUE HC Series node provides an interlocking set of options that let you create a comprehensive security strategy for the node.

Refer to the GigaVUE-OS Administration Guide for detailed information.

Refer to the following sections for configuration examples.

IP Filter Chains for Security

An IP filter is a chain of rules for the treatment of packets. Chains can be INPUT, OUTPUT, or FORWARD. Chains have a policy (or default target) of either ACCEPT or DROP. The policy is applied to a packet if it reaches the end of the chain. Each rule in the chain specifies the packets that match.

NOTE: Use the **ip filter chain INPUT policy DROP** command with caution, as it can make the device unreachable if not properly configured. Before enabling this policy, ensure that you have added the necessary ACCEPT rules using the CLI command **ip**

filter chain OUTPUT rule append tail target ACCEPT <>, where <> represents the required IP addresses, ports, and protocols. To verify that the rules are correctly configured, run **show ip filter**, and then use **ip filter enable** to apply the changes.

The following configuration examples use the **ip filter chain** and **ipv6 filter chain** commands. For details on these commands, refer to ip and ipv6 in the reference section.

For examples of using IP filter chains, refer to the following:

- Close Open Ports
- Management Port Security
- NTP Server Security
- Allowing IGMP Traffic

NOTE: The IP filtering capabilities of these CLI commands are provided by iptables and ip6tables in the Linux kernel.

Close Open Ports

With the exception of ports used for Web and SSH, ports are normally closed. The following configuration example closes ports that may be open.

To close open ports, execute the following commands:

Step	Description	Command
1.	Configure a rule for the chain and specify the destination port numbers. Append tail adds a new rule after all existing rules. Dup-delete specifies that after adding a rule, delete all other existing rules that are duplicates of it.	**(config) # ip filter chain INPUT rule append tail target DROP dup-delete dest-port 256 in-intf eth0 protocol udp** **(config) # ip filter chain INPUT rule append tail target DROP dup-delete dest-port 512 in-intf eth0 protocol udp** **(config) # ip filter chain INPUT rule append tail target DROP dup-delete dest-port 111 in-intf eth0 protocol udp**
2.	Enable IP filtering.	**(config) # ip filter enable**
3.	Display IP filter configuration	**(config) # show ip filter**

Management Port Security

Management port security lets you restrict the exchange of packets through the management port by creating an access control list to restrict user and SNMP access.

NOTE: Exercise caution when using the following configuration example so as not to interfere with communications through the backplane or within a cluster.

Getting Started

It is recommended that you connect to the console as follows:

(config) # **serial baudrate 115200**

(config) # **serial enable**

During configuration, you may not be able to access the node through SSH.

If for any reason, you are locked out of the node, execute the following commands:

(config) # **ip filter chain INPUT policy ACCEPT**

(config) # **no ip filter enable**

Configure Management Port Security

In this sample configuration, there are five unique ports to access the host so there are five IP addresses you want available to the node. This sample configuration configures an INPUT chain with a DROP policy, which means that any incoming packets will be rejected unless they have the IP source addresses specified in the INPUT chain rule (any packets that do not match a rule in the INPUT chain are dropped).

NOTE: Once you execute the **ip filter enable** command, only the configured IP addresses will have access.

To implement management port security, execute the following commands:

Step	Description	Command
1.	Configure a rule for the chain and specify the first to the fifth source IP addresses. (Append tail adds a new rule after all existing rules.)	**(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.131 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.132 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.133 255.255.255.255**

Step	Description	Command
		(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.134 255.255.255.255
2.	Configure a policy for the chain. DROP means that any packets not matching a rule in the INPUT chain will be dropped.	**(config) # ip filter chain INPUT policy DROP**
3.	Permit GigaSMART card communication for the out-of-band cluster.	**(config) # ip filter chain INPUT rule append tail target ACCEPT dest-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255**
4.	Configure the IP addresses of the permitted hosts (cluster nodes).	**(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.50.22.130 255.255.255.255**
5.	Configure a policy for the chain. DROP means that any packets not matching a rule in the INPUT chain will be dropped.	**(config) # ip filter chain INPUT policy DROP**
6.	Enable IP filtering.	**(config) # ip filter enable**
7.	Display IP filter configuration	**(config) # show ip filter**

Displaying Management Port Security

Use the following command to display the IP filter chain:

(config) # show ip filter

```
Packet filtering for IPv4: enabled
```

```
Apply filters to bridges: no
Active IPv4 filtering rules (omitting any not from configuration):
Chain 'INPUT'
  #  Target  Proto   Source              Destination         Other
  1  ACCEPT  icmp    all                 all
  2  ACCEPT  all     all                 all                 inb lo
  3  ACCEPT  all     10.50.22.130/32     all
  4  ACCEPT  all     10.50.22.131/32     all
Policy: DROP
Chain 'OUTPUT'
 No rules.
Policy: ACCEPT
Chain 'FORWARD'
 No rules.
Policy: DROP
```

This sample configuration does not affect any output packets (such as those generated by a trap). However, any packets coming from an NTP server, for example, will be blocked unless the IP address is added to the chain (as a source address). Refer to NTP Server Security for another configuration example.

NTP Server Security

With NTP server security, you can permit access to the NTP server.

NOTE: Exercise caution when using the following configuration example so as not to interfere with communications through the backplane or within a cluster.

Configure NTP Server Security

This sample configuration is for an out-of-band cluster environment. You configure the management IP addresses of the permitted hosts or cluster nodes and the workstation hosts that are permitted to access the nodes.

To implement NTP server security, execute the following commands:

Step	Description	Command
1.	Permit an NTP server by specifying the NTP port number and IP address of an Internet time server.	**(config) # ip filter chain INPUT rule append tail target ACCEPT protocol udp source-port 123 source-addr 129.6.15.0 /24**
2.	Permit multicast Domain Name System (mDNS) for the out-of-band cluster.	**(config) # ip filter chain INPUT rule append tail target ACCEPT dest-addr 224.0.0.251 /32**
3.	Permit GigaSMART card	**(config) # ip filter chain INPUT rule append tail**

Step	Description	Command
	communication for the out-of-band cluster.	**target ACCEPT dest-addr 12.19.148.0 /24** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 12.19.148.0 /24**
4.	Configure the IP addresses of the permitted hosts (cluster nodes).	**(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.79 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.80 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.81 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.82 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.83 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.25.84 /32**
5.	Configure the IP addresses of the permitted workstation hosts.	**(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.40.21.140 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.40.21.85 /32** **(config) # ip filter chain INPUT rule append tail target ACCEPT source-addr 10.115.122.36 /32**
6.	Configure a policy for the chain. DROP means that any packets not matching a rule in the INPUT chain will be dropped.	**(config) # ip filter chain INPUT policy DROP**
7.	Enable IP filtering.	**(config) # ip filter enable**
8.	Display IP filter configuration	**(config) # show ip filter**

Allowing IGMP Traffic

IP filter chains can also be used to allow IGMP protocol traffic in a clustering environment. Refer to the *"Best Practices for OOB Clusters with IGMP Snooping"* section in the *GigaVUE Fabric Management Guide* for details.

Disable a Serial Console Port

For security reasons, it may be necessary at times to disable a serial console port.

WARNING: Before you disable a serial console port, make sure you have Web (HTTPS) or SSH connections in order to enable the port at a later time.

The following example shows the CLI command to disable a serial console port:

```
(config) # no serial ?enable
 Disable serial console access
(config) # no serial enable
```

The following confirmation message displays:

```
Disable serial console will make serial connection unusable.
Only use this config command when you have available ssh connections.
Enter 'YES' to confirm this operation:
```

Typing YES displays the following message:

```
Enter 'YES' to confirm this operation: YES
Serial Console disabled.
```

Configure Role-Based Access: A Summary

Configuring role-based access consists of the major steps listed in the following table:

Step	Description
Configure Roles	Administrators use the **aaa authorization roles role <role name> description <description>** command to create roles. At first, roles are empty containers. You can create as many as you need to share the Deep Observability Pipeline effectively. For example, if you have an IT organization with six different groups (Security, Desktop, Application Performance Management, Server, Archive, and so on), each with different packet needs, you may want to create separate roles for each of them and assign them to different sets of tool ports. **Note**: The built-in "Default" role has no access to unassigned ports.
Create Users with Roles Assigned	Once you have roles created, you can assign them to users. You can assign roles to existing users or as you create new users. Users can have multiple roles assigned, giving them access to different sets of ports. Use the **username** command to assign roles. The syntax is as follows: **(config) # username <username> roles [add | replace] <roles>** As with most CLI commands, you can preface this command with **no** to remove selected roles from a specified users. Keep in mind that admin-level users automatically have access to all roles. Administrators assign roles to default-level users.
Associate Roles with Ports and Permissions	The final step is to associate roles with ports and permissions. A user with a particular role will have access to all ports assigned that role at the designated permission level. Use the **port** command to associate roles with ports and permissions. The syntax is as follows: **(config) # port <port> assign role <role> level <level>** As with most CLI commands, you can preface this command with **no** to remove selected roles from a specified port.

Step	Description
Restriction for Removing a Role	An error message is displayed if you try to remove a role when it is used in a port tool-share. For example: **(config) # aaa authorization roles role role2** **(config) # port 1/3/x2 type tool** **(config) # port 1/3/x3 type tool** **(config) # port 1/3/x2 tool-share role role2** **(config) # port 1/3/x3 tool-share role role2** **(config) # no aaa authorization roles role role2** `% Role role2 is used in the port 1/3/x2,1/3/x3 tool-share.` To remove the role, use the following CLI commands: **(config) # no port 1/3/x2..x3 tool-share role role2** **(config) # no aaa authorization roles role role2**
Fine Tune and Evolve	The Deep Observability Pipeline evolves as your needs change. You can continue to add new roles and tweak assigned ports and permissions to achieve the sharing results needed for different groups to get the packets they need.

Role-Based Access: Required Permissions by Command

The following table summarizes the minimum rights required to perform different tasks on the GigaVUE-OS® HC Series node.

Note the following:

- Tasks that can only be performed by the built-in admin role are listed as **admin**.
- Tasks that can be performed by a non-admin user are listed with the minimum permission level required to perform the task – for example, **Level 2+** indicates that a user with either Level 2, Level 3, or Level 4 permissions on the port can perform the task.
- When performing tasks on a **<port-list>**, the access granted is the lowest level among all the ports in the list. For example, if the user **jhalladay** has **Level 2** rights on **2/3/x4** and Level 3 rights on **3/4/x6..x8**, the system would not allow **jhalladay** to perform a task requiring **Level 3** permissions on a port-list containing both **2/3/x4** and **3/4/x6..x8**. However, **jhalladay** would be able to perform tasks requiring Level 3 permissions on a port-list containingLevel 2+ just **3/4/x6..x8**.

Component	Show	Create	Delete	Modify
port	Level 1+	admin	admin	Port Attributes: ▪ port type – admin and Level 4 ▪ params (such as, admin,

Component	Show	Create	Delete	Modify
				speed) – Level 3 ▪ timestamp – Level 2+ ▪ lock/lock-share – Level 2+
tool port-filter	Level 1+	Level 2+	Level 2+	N/A
port-pair Indicated permissions required on both source and destination of port-pair.	Level 1+	Level 3	Level 3	Link Failure Propagation (lfp) – Level 3
port-group (all ports in group)	Level 1+	Level 2+	Level 2+	Modifications to a port-group's ports require Level 2+ permissions on both the existing ports in the group and any new ports to be added.
gigastream (all ports in GigaStream)	Level 1+	Level 2+	Level 2+	Modifications to a GigaStream's ports require Level 2+ permissions on both the existing ports in the GigaStream and any new ports to be added.
gsgroup(all GigaSMART engine ports in GigaSMART group)	Level 1+	Level 2+	Level 2+	Modifications to a GigaSMART group's ports require Level 2+ permissions on both the existing ports in the GigaSMART group and any new ports to be added. GigaSMART groups consist of one or more of the GigaSMART engine ports on a GigaSMART line card or module on GigaVUE-OS HD Series, or GigaVUE-HC3, or GigaVUE-HB1 or GigaVUE-HC1 node. They provide the processing power for GigaSMART operations.
gsopRequires the indicated permission level on all GigaSMART engine ports in the GigaSMART group to which the gsop is assigned for processing.	Level 1+	Level 2+	Level 2+	Modifications to a GigaSMART operation (gsop) require Level 2+ permissions on all of the GigaSMART engine ports in the GigaSMART group to which the gsop is assigned for processing. This means that a user must have Level 2+ permissions on a GSOP's GigaSMART engine ports in order to use it in a map.
tool-mirror Indicated permissions required on both source and destination of port-	Level 1+	Level 2+	Level 2+	Modify comment – Level 2+

Component	Show	Create	Delete	Modify
pair.				
map Indicated permission required on network, inline-network, tool, inline-tool, or collector ports.	Level 1+ on at least one of the network ports.	Level 2+	Level 2+	Any changes to a map's ports (network, inline-network, tool, inline-tool, or collector), rules, or GigaSMART operations require Level 2+ permissions on all existing and proposed new ports in the map.
inline-network, inline-tool	Level 1+	Level 2+	Level 2+	Modifications to inline-network and inline-tool ports require Level 2+ permissions on both the port pairs.

Role-Based Access: Rules and Notes

Refer to the GigaVUE-OS Administration Guide for User Management and Role Management details.

Port Ownership

- Only administrators can assign or remove roles from ports.
- To remove a user's lock from a port, administrators use the following command:

  (config) # no port <port-list> lock
- To remove a user's lock-share, administrators use the following command:

  (config) # no port <port-list> lock-share user <username>
- Administrators can also lock a port for a user. For example, the following command locks port 1/1/g1 for user **psandoval**.

  (config) # port 1/1/g1 lock user psandoval
- The admin role automatically has Level 4 permissions to all ports. The admin role cannot be assigned to any port.

Traffic Configuration

Refer to the Role-Based Access: Required Permissions by Command section summarizes the required permissions for traffic-related commands.

CLI Commands for Role-Based Access

The main commands for role-based access are summarized in the following table:

CLI Commands for Role-Based Access
show usernames
Reviewing User and Role Assignments

CLI Commands for Role-Based Access
show usernames assignment <all | alias> // show user assignments, including roles, locks, and lock-shares
show role assignment <all | alias> // show role's users, assigned ports, description
show port assignment <all, box-id, port-list, slot> // show the roles assigned to a port at each permission level
show port access <all, box-id, slot> // show the roles that can access a port, including any locks and lock-shares in place
Specifying Authentication Methods and Order
aaa authentication login default [list of authentication methods] // authentication methods order (refer to *Configuring AAA*).
aaa authorization map order <local-only | remote-first | remote-only> // change authorization mode (refer *Configuring AAA*)
aaa authorization map default-user <local-user-name> // default mapped user
Creating and Removing Roles
[no] aaa authorization roles role <role_name> // define new role
[no] aaa authorization roles role <role_name> description "role" // define new role with description
Assigning and Removing Roles for Users
[no] username <user_name> roles add <roles separate by space(s)> // Assign roles to an user
no username <user_name> roles all // Remove all user's roles, except the Default role
username <user-name> roles replace <roles separate by space(s)> // Replace current role-set with new role-set
Assigning and Removing Roles and Locks from Ports
[no] port <ids> assign role <role_name> [level 1|2|3] // Assign role to port, default is 1
no port <ids> assign role all // Remove all assigned roles from input port(s)
[no] port <ids> lock // Lock a port(s)
[no] port <ids> lock user <username> // Administrator uses to lock ports for another user
[no] port <ids> lock-share user <user name> // Lock owner can use this to share access to port at sharer's permission level.
no port <ids> lock-share all // remove all lock-shares

CLI Commands for Role-Based Access
[no] port <ids> tool-share role <role> // Assign or remove roles from a port's tool share list.
Enabling Extra Roles in AAA Servers
[no] tacacs-server extra-user-params roles enable // enable extra roles (refer to *Configuring AAA*)
[no] radius-server extra-user-params roles enable
[no] ldap extra-user-params roles enable

Admin-Only CLI Commands

The following commands are only available to admin users:

Admin-Only CLI Commands	
aaa accounting changes default *	no snmp-server community *
aaa authentication attempts class-override *	no snmp-server enable *
aaa authentication attempts lockout *	no snmp-server user *
aaa authentication attempts track enable	no ssh client *
aaa authentication login *	ssh client global *
aaa authorization map *	
aaa authorization roles *	[no]ssh server enable
	ssh server host-key *
no aaa *	ssh server ports *
configuration * (except for "configuration write")	reset factory *
	show aaa authentication attempts status user <username>
[no] image *	show snmp user
[no] ldap *	uboot install
[no] boot *	
[no] radius-server *	[no] card *
[no] tacacs-server *	[no] chassis box-id *
	[no] port <port-id> assign *
snmp-server community <community>	chassis migrate *
snmp-server enable communities	no traffic all
snmp-server enable mult-communities	
	[no] clock

Admin-Only CLI Commands	
[no] cluster	**halt**
[no] hostname	**[no] interface**
[no] ip	**[no] ipv6**
[no] ntp	**ntpdate**
[no] ptp	**reload**
reset	**[no] web**

Configure AAA

Use the **aaa** command in Configure mode for **authentication**, **authorization**, and **accounting** settings for the GigaVUE-OS® HC Series node – there are separate arguments for each. In general, configuring authentication consists of specifying the login methods accepted, the order in which they are tried, the local user account to map to external logins, whether to accept roles specified by the AAA server, and the configuration of the external authentication server itself.

Refer to the GigaVUE Fabric Management Guide for detailed information.

Configure AAA Authorization

For details on the **aaa authorization** command, refer to aaa authorization.

Example

The following commands demonstrate how to set up authentication using RADIUS with a fallback to local if no RADIUS server is available.

Command	Description
(config) # aaa authentication login default radius local	Use RADIUS authentication first, followed by local authentication.
(config) # aaa authorization map order remote-first	If the external user also exists in the local database, use the specified local account. Otherwise, use the account specified by the **map default-user** argument.
(config) # aaa authorization map default-user admin	If the external user does not exist in the local database, use the **admin** account instead. This is only done if the **map order** is set to **remote-first** or **local**.
(config) # radius-server host 192.168.0.62 key gigamon	Adds a RADIUS server at IPv4 address **192.168.0.62** to the GigaVUE HC Series node's list.
(config) # radius extra-user-params roles enable	Allows the RADIUS server to include additional roles for a

Command	Description
	remotely authenticated user in the response. Refer to *Granting Roles with External Authentication Servers* on page 935.

Add a RADIUS Server

Admin users use the radius-server command to specify the RADIUS servers to be used for authentication. You can specify multiple RADIUS servers. Servers are used as fallbacks in the same order in which they are specified. If the first server is unreachable, the second is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

For details on the **radius-server** command, refer to radius-server.

Examples

The following **radius-server** commands demonstrate different ways to specify a RADIUS server:

Command	Comments
(config) # radius-server host 192.168.0.75 key gvhd8 **(config) # radius-server timeout 20** **(config) # radius extra-user-params roles enable**	Specifies that: • Users logging in through RADIUS will be authenticated against the RADIUS server at IPv4 address **192.168.0.75**. • Authentication packets will be encrypted using the string **gvhd8**. • The default value of 1812 will be used for the **auth-port**. • The global values for **retransmit** and **timeout** will be used because they are not explicitly specified in the **host** command. • The second command changes the global timeout setting to 20 seconds. • The third command specifies that extra roles sent in the response from the AAA server will be honored if they match an existing role configured locally on the node.
(config) # radius-server host 192.168.1.212 auth-port 5150 key lowkey retransmit 5 timeout 30	Specifies that: • Users logging in through RADIUS will be authenticated against the RADIUS server at IPv4 address **192.168.1.212**. • Authentication packets will be encrypted using the string **lowkey**. • The non-standard authentication port of 5150 will be

Command	Comments
	used. • The global **retransmit** and **timeout** values will be overridden with the per-host values specified here. **Note**: If this command was used after the command in the previous row, this server would be the backup RADIUS server for the previously-specified server.
(config) # radius-server host www.MyCo.com	Specifies a RADIUS server host by hostname. **Note**: Starting in software version 5.6, GigaVUE-OS supports dynamic Fully Qualified Domain Name (FQDN) or hostname.

Delete a RADIUS Server

As with all GigaVUE-OS elements, you can delete a RADIUS server by prefacing the **radius-server** command with the **no** command. For example, the following command deletes the second RADIUS server:

(config) # no radius-server host 192.168.1.212

The following command resets the global **timeout** value to its default setting of three:

(config) # no radius-server timeout

The **show radius** output looks as follows after making these changes:

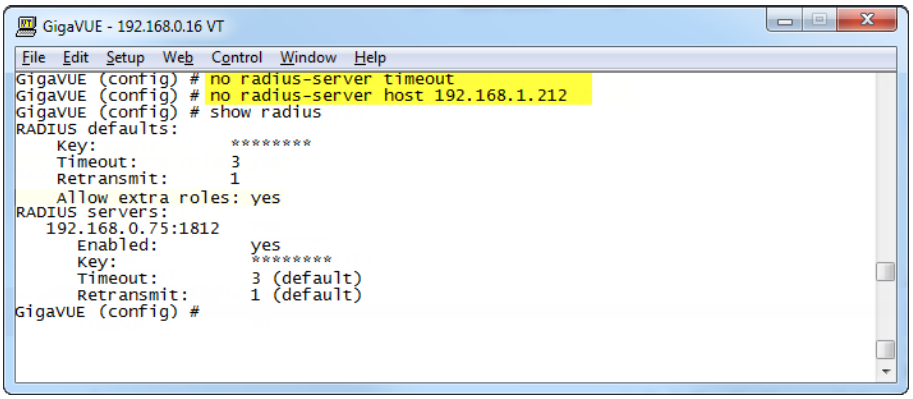


Figure 28 *show radius Output after Removing Server*

Add a TACACS+ Server

Admin users use the **tacacs-server** command to specify the TACACS+ servers to be used for authentication. You can specify multiple TACACS+ servers. Servers are used as fallbacks in the same order in which they are specified. If the first server is unreachable, the second is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

For details on the parameters and the syntax of the **tacacs-server** command, refer to tacacs-server in the reference section.

Examples

The following **tacacs-server** commands demonstrate different ways to specify a TACACS+ server:

Command	Comments
(config) # tacacs-server host 192.168.0.93 key mytac123 **(config) # tacacs-server retransmit 5** **(config) # tacacs-server extra-user-params roles enable** **service gigamon**	Specifies that: • Users logging in through TACACS+ will be authenticated against the TACACS+ server at IPv4 address **192.168.0.93**. • Authentication packets will be encrypted using the string **mytac123**. • The default value of 49 will be used for the **auth-port**; the default value of **pap** will be used for the auth-type. • The global values for **retransmit** and **timeout** will be used because they are not explicitly specified in the **host** command. • The second command changes the global **retransmit** setting to 5 attempts. • The third command specifies that extra roles sent in the response from the TACACS+ server will be honored if they match an existing role configured locally on the node. • The last command sets the authorization service to **gigamon** for successful integration with Cisco ACS 5.x.
(config) # tacacs-server host 2001:db8:a0b:12f0::11:49 auth-port 4949 key tackey retransmit 3 timeout 45	Specifies that: • Users logging in through TACACS+ will be authenticated against the TACACS+ server at IPv6 address **2001:db8:a0b:12f0::11:49**. • Authentication packets will be encrypted using the string **tackey**. • The non-standard port **4949** will be used instead of 49. • The global **retransmit** and **timeout** values will be overridden with the per-host values specified here (**3** and **45**, respectively). **Note**: If this command was used after the command in the previous row, this server would be the backup TACACS+ server for the previously-specified server.
(config) # tacacs-server host www.MyCo.com	Specifies a TACACS+ server host by hostname. **Note**: Starting in software version 5.6, GigaVUE-OS supports dynamic Fully Qualified Domain Name (FQDN) or hostname.

Figure 2 shows the results of a **show tacacs** command for the servers set up in the first two examples:

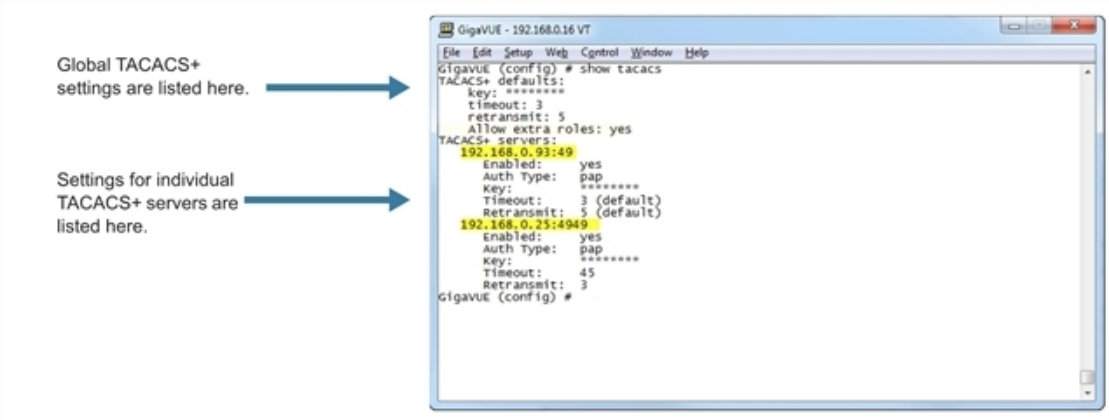


Figure 29 *Results of show tacacs Command*

Delete a TACACS+ Server

As with all GigaVUE-OS elements, you can delete a TACACS+ server by prefacing the **tacacs-server** command with the **no** command. For example, the following command deletes the second TACACS+ server:

(config) # no tacacs-server host 192.168.0.25

The following command resets the global **retransmit** value to its default setting of one:

(config) # no tacacs-server retransmit

The **show tacacs** output looks as follows after making these changes:

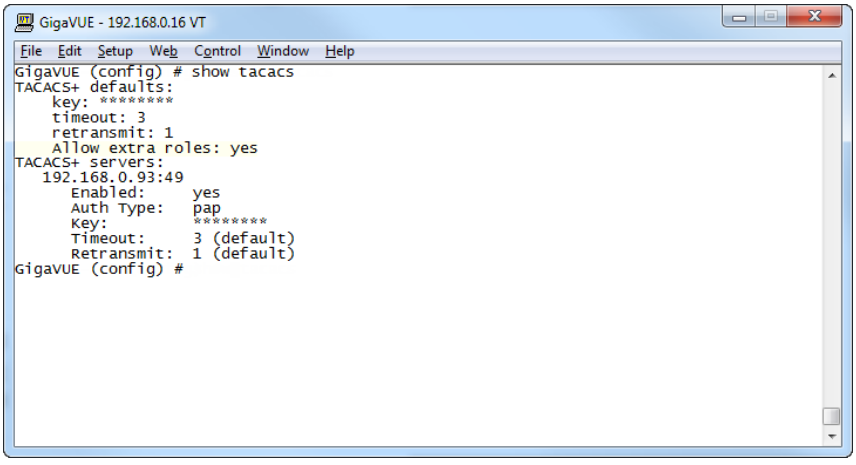


Figure 30 *show tacacs Output after Removing Server*

Configure an IPv6 Address

Use the following CLI command to configure an IPv6 address for a TACACS+ server:

(config) # tacacs-server host 2001:db8:a0b:12f0::17/120 key gigamon enable

To enable IPv6 on the GigaVUE-OS node, there are more configuration steps. Refer to IPv6 Configuration Example.

Add an LDAP Server

Admin users use the **ldap** command to specify the LDAP servers to be used for authentication. You can specify multiple LDAP servers. Servers are used as fallbacks in the same order in which they are specified. If the first server is unreachable, the second is tried, and so on, until all named servers have been used. If a server is reachable and authentication fails, the authentication process terminates.

For Common Criteria, specify SHA password hashing when configuring the remote LDAP server. For details on Common Criteria, refer to *Common Criteria* on page 962.

For details on the **ldap** command, refer to *ldap* on page 239.

Examples

The following highlights the steps to configure LDAP, AAA, and certificates.

General configuration:

```
username operator password <enter password>
DNS IP is configured <DNS/LDAP server IP>
```

LDAP configuration:

```
ldap bind-dn <accountname@domainname>
ldap bind-password <account password>
ldap login-attribute <AD: sAMAccountName or openldap: uid>
ldap base-dn cn=Users,dc=example,dc=com (directory tree search path)
ldap port 636 (ssl: 636, none-ssl: 389)
ldap host <dns name of server> (ip address will not work)
In environments where the device is isolated from DNS, use the following:
ip host <host>.<domain>.<com> x.x.x.x
```

AAA configuration:

```
aaa authentication login default ldap local
aaa authorization map order remote-first
```

Certificate creation:

On Linux or Linux app (such as Cygwin):

```
$ openssl.exe req \> -x509 -nodes -days 365 \> -newkey rsa:1024 -keyout mycert.pem -out
mycert.pem
Generating a 1024 bit RSA private
key.....................................++++++.......++++++writing new private key to
'mycert.pem'-----You are about to be asked to enter information that will be incorporated
into your certificate request.What you are about to enter is what is called a Distinguished
```

```
Name or a DN.There are quite a few fields but you can leave some blankFor some fields there
will be a default value,If you enter '.', the field will be left blank.-----Country Name (2
letter code) [AU]:USState or Province Name (full name) [Some-State]:CaliforniaLocality Name
(eg, city) []:OurtownOrganization Name (eg, company) [Internet Widgits Pty
Ltd]:OurcompanyOrganizational Unit Name (eg, section) []:OurGroupCommon Name (e.g. server
FQDN or YOUR name) []:MynameEmail Address []:My.name@Ourcompany.com
$ ls
mycert.pem
```

Certificate installation:

- crypto certificate name <name> public-cert pem “<certificate string including BEGIN CERTIFICATE and END CERTIFICATE lines>”. Enclose the contents of the PEM file in quotation marks. Refer to the example.

  NOTE: Install all the certificates in the certificate chain.

- crypto certificate name <name> public-cert pem fetch <url>

  NOTE: Fetch the remote certificate using SCP, HTTP, HTTPS, TFTP, and SFTP.

- crypto certificate ca-list default-ca-list name <installed certificate>

  NOTE: Execute this for all the installed certificates.

- sh crypto certificate ca-list default-ca-list <this command shows all the installed certificate>

Example:

```
(config) # crypto certificate name mycert public-cert pem
"-----BEGIN CERTIFICATE-----

MIIC8jCCAlugAwIBAgIJAJtFtchQpGk6MA0GCSqGSIb3DQEBBQUAMIGRMQswCQYDVQQGEwJVUzETMBEGA1UECAwKQ2Fs
aWZvcm5pYTERMA8GA1UEBwwITWlscGl0YXMxEDAOBgNVBAoMB0dpZ2Ftb24xCzAJBgNVBAsMAlFBMRAwDgYDVQQDDAdi
ZXJuYXJkMSkwJwYJKoZIhvcNAQkBFhpiZXJuYXJkLmFydG9sYUBnaWdhbW9uLmNvbTAeFw0x

MzEwMTcxOTQzNDJaFw0xNDEwMTcxOTQzNDJaMIGRMQswCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTERMA8G
A1UEBwwITWlscGl0YXMxEDAOBgNVBAoMB0dpZ2Ftb24xCzAJBgNVBAsMAlFBMRAwDgYDVQQDDAdiZXJuYXJkMSkwJwYJ
KoZIhvcNAQkBFhpiZXJuYXJkLmFydG9sYUBnaWdhbW9uLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwSM5
dHUaZUcI8vTdd+l/I+2dXxamSFI2xLLS54WunKaLfI9Fm6FS6NYzoPY7SAS+Y5qtsFR5di+duPhpylcDTCDUBa0CMzdt
zt0qGR3uuxClNWt6cBKFaLGMwqgxe+XAtqt5S5FzEXZGZp9bmuwpLhpXm7Dhhkfa+YjkzHhbeoECAwEAAaNQME4wHQYD
VR0OBBYEFGB4M/57N9yDBT3ODiUV4r/Evk6BMB8GA1UdIwQYMBaAFGB4M/57N9yDBT3ODiUV4r/Evk6BMAwGA1UdEwQF
MAMBAf8wDQYJKoZIhvcNAQEFBQADgYEAYlasOq3/oB8Yu7Y44NZnbrhrWUZZleYNLa3c8+8KnSvVZYsUZJBUXNthOs3n
vlRW+Z8H9J1D9PkC/a5ym2Na3AU0zXpTt7HQA0cfemKJqJ7XIF/7AUOJSjIxxLMQL+2tGkc5on8No27wn5UgLFzbn9Zz
U/QDkD3eKOvKtQW50Ok=
-----END CERTIFICATE-----"
```

Successfully installed certificate with name 'mycert'

Examples

The following **ldap** commands demonstrate different ways to add an LDAP server to the GigaVUE HC Series node’s list:

Command	Comments
(config) # ldap host 192.168.0.62 **(config) # ldap login-attribute uid** **(config) # ldap base-dn "ou=People,dc=ncgold,dc=com"** **(config) # ldap extra-user-params roles enable**	Specifies that: • Users logging in through LDAP will be authenticated against the LDAP server at IPv4 address **192.168.0.62**. • The login name sent from the GigaVUE HC Series node will match the User ID in the LDAP server. • The base distinguished name of the user information in the LDAP server's schema is **ou=People,dc=ncgold,dc=com**. That is, the user ID will be found in the People organizational unit of the ncgold.com DN. • Extra roles sent in the response from the LDAP server will be honored if they match an existing role configured locally on the node. • Default values will be used for all other settings.
(config) # ldap host 2001:10:115:104::110:f **(config) # ldap port 5858** **(config) # ldap base-dn "ou=Employees,dc=ncgold,dc=com"** **(config) # ldap timeout 35** **(config) # ldap version 2**	Specifies that: • Users logging in through LDAP will be authenticated against the LDAP server at IPv6 address **2001:10:115:104::110:f**. • The base distinguished name of the user information in the LDAP server's schema is **ou=Employees,dc=ncgold,dc=com**. That is, the user ID will be found in the Employees organizational unit of the ncgold.com DN. • The non-standard port **5858** will be used instead of 389. This is a global setting and not a per-host setting as it is for other AAA servers. • The GigaVUE HC Seriesnode will use LDAPv2 instead of v3. • The global **timeout** values is now set to 35 seconds. **Note**: If this command was used after the command in the previous row, this server would be the backup LDAP server for the previously-specified server. You can always change the order of LDAP servers by using the **ldap host <host> order** command.
(config) # ldap host 192.168.1.84 order 1	This command moves the LDAP server at IPv4 address **192.168.1.84** up to the first position in the list of the servers, meaning that it will be used for authentication first.
(config) # ldap host www.MyCo.com	Specifies an LDAP server host by hostname. **Note**: Starting in software version 5.6, GigaVUE-OS supports dynamic Fully Qualified Domain Name (FQDN) or hostname.

Figure 4 shows the results of a **show ldap** command for the servers set up in the first two examples. Note that most LDAP settings are global settings – changes made for the second server appear as new settings for the defaults.

Figure 31 *Results of show ldap Command*

Delete an LDAP Server

As with all GigaVUE-OS elements, you can delete an LDAP server by prefacing the **ldap host** command with the **no** command. For example, the following command deletes the second ldap server:

(config) # no ldap host 192.168.1.84

The following commands reset the LDAP port, timeout, and version back to their default settings:

(config) # no ldap port

(config) # no ldap timeout

(config) # no ldap version

The **show ldap** output looks as follows after making these changes:

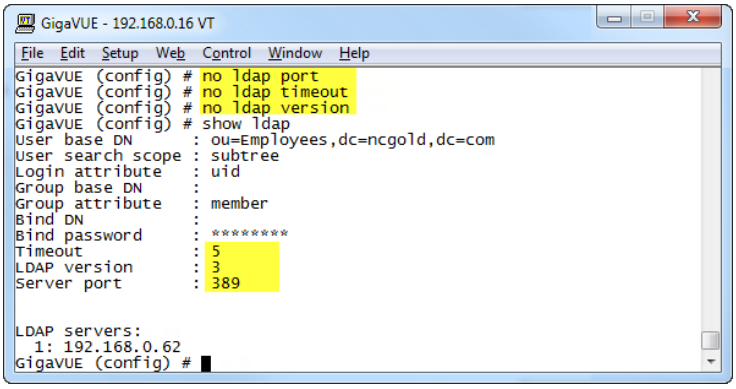


Figure 32 *show ldap Output after Resetting Values to Defaults*

Configuring an IPv6 Address

Use the following CLI command to configure an IPv6 address for an LDAP server:

(config) # ldap host 2001:db8:a0b:12f0::66

To enable IPv6 on the GigaVUE-OS node, there are more configuration steps. Refer to IPv6 Configuration Example.

IPv6 Configuration Example

The following example configures and enables an IPv6 address on a GigaVUE-HC3 node. It also configures an IPv6 address for a TACACS+ server, which is the remote authenticator.

To configure an IPv6 address for an LDAP server, use the CLI command **host ldap host** instead of **tacacs-server host** in Step 13.

In this example, the IPv6 network prefix is 2001:db8::/120.

Step	Description	Command
1.	Use the configuration jump-start wizard on the GigaVUE-HC3 to specify the eth0 Management interface, enable DHCP on eth0, enable IPv6, enable IPv6 autoconfig on eth0, and enable DHCPv6 on eth0.	**(config) # configuration jump-startGigaVUE-OS configuration wizardStep 1: Hostname? [gigamon1]Step 2: Management Interface <eth0> ? [eth0]Step 3: Use DHCP on eth0 interface? [no] yesStep 4: Enable IPv6? [no] yesStep 5: Enable IPv6 autoconfig (SLAAC) on eth0 interface? [no] yesStep 6: Enable DHCPv6 on eth0 interface? [no] yesStep 7: Enable secure cryptography? [no]Step 8: Enable secure passwords? [no]Step 9: Minimum password length? [8]Step 10: Admin password?Please enter a password. Password is a must.Step 10: Admin password?Step 10: Confirm admin password?Step 11: Cluster enable? [no]You have entered the following information:To change an answer, enter the step number to return to.Otherwise hit <enter> to save changes and exit.<enter>**
2.	Verify that IPv6 is supported and enabled. **NOTE:** If **no** is displayed for either IPv6 supported or admin enabled, IPv6 communication will fail. Issue the CLI command in Step 3.	**(config) # show ipv6** `IPv6 summary  IPv6 supported: yes  IPv6 admin enabled: yes  IPv6 interface count: 1`

Step	Description	Command
3.	(Optional) If **no** is displayed for either IPv6 supported or admin enabled, issue the following CLI command.	**(config) # ipv6 enable**
4.	Configure an IPv6 address for eth0 in the same subnet as the TACACS+ server.	**(config) # interface eth0 ipv6 address 2001:db8:a0b:12f0::17/120**
5.	Enable the eth0 interface.	**(config) # interface eth0 ipv6 enable**

6.	Verify the IPv6 address.	**(config) # show interface eth0**
		`Interface eth0 status:... IPv6 enabled: yes ...`
		`IPv6 address: 2001:db8:a0b:12f0::17/120`

7.	Add an IPv6 static route. Routing is essential to IPv6 communications. Ensure that all IPv6 addresses are routable and are on the same subnet. Also ensure your network switches, routers, and firewalls are configured in such way to allow IPv6 packets to reach their destination.	**(config) # ipv6 route 2001:db8:a0b::/120 eth0**
8.	Verify the IPv6 routing table.	**(config) # show ipv6 route**

```
                        Destination prefix
                            Gateway                                     Interface  Source
                        ------------------------------------------------------------------
  --
                        ::/0
                            ::                                          eth0       static
                        2001:db8:a0b:12f0::/120
                            ::                                          eth0       static
                            ::                                          eth0
  interface
                        2001:db8:a0b:12f0::17/120
                            ::                                          lo         local
```

Step	Description	Command
9.	Ping the host default gateway from the GigaVUE-OS node to verify the endpoints. If you do not get a response, check the connections and routing.	**(config) # ping6 -I eth0 2001:db8:a0b:12f0::1**
10.	Configure TACACS+ as the default AAA login.	**(config) # aaa authentication login default tacacs+**
11.	Verify the authentication configuration. Also verify that the default user is external.	**(config) # show aaa** AAA authorization: Default User: **external** Map Order: remote-first Authentication method(s): **tacacs+** local
12.	Verify that the external user has a password set.	**(config) # show usernames** USERNAME FULL NAME ACCOUNT STATUS admin System Administrator **Password set**
13.	Configure the IPv6 address for the TACACS+ server.	**(config) # tacacs-server host 2001:db8:a0b:12f0::11 key gigamon enable**
14.	Verify the TACACS+ server IPv6 address.	**(config) # show tacacs** ... TACACS+ servers: **2001:db8:a0b:12f0::11:49**
15.	Ping the TACACS+ server to verify that it is reachable.	**(config) # ping6 -I eth0 2001:db8:a0b:12f0::11**

Encrypt Syslog Audit Data

Syslog audit data, such as messages and traps, are usually sent unencrypted between a GigaVUE-OS node and the syslog server using UDP over port 514. The messages are sent in plain text. To allow secure transmission, starting in software version 4.4, you can send encrypted syslog audit data by using TCP and SSH options.

For more information on the CLI commands used in the following section, refer to logging and ssh in the reference section.

Encryption Procedure

Use the following sample procedure to encrypt syslog audit data.

Step	Description	Command
Configure TCP Port		
1.	On the GigaVUE-OS node, configure the TCP port on which the syslog server listens. **Note**: Use the **logging** command to specify an IPv4 address, an IPv6 address, or a hostname.	**(config) # logging 192.168.1.25 tcp 51300**
Generate Public Key		
2.	On the GigaVUE-OS node, generate a public key for a user account on the node. If the user has the admin role, you must use the default **admin** account to generate the public key.	**(config) # ssh client user admin identity rsa2 generate**
3.	On the GigaVUE-OS node, display the key contents, then copy the key contents.	**(config) # show ssh client** For example, copy all of the key contents: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCZfGVW4+2S9Lng1Jo5lM7jJdnU93N4i ... grneMozmTIvJgB3vfV
Login to Server to Paste the Key		

Step	Description	Command
4.	In Linux, log into the syslog server. In the following example, the IP address of the syslog server is 192.168.1.25 and the username is an existing user called sysloguser.	Log into IP address 192.168.1.25 with username sysloguser to see the Linux prompt: #
5.	On the syslog server, change the directory to .ssh.	**# cd .ssh**
6.	On the syslog server, edit the authorized_keys file, located in the .ssh directory, using any editor, then paste the key contents.	For example, using the vi editor: # vi authorized_keys Paste the key contents that you copied in the previous step into the authorized_keys file in the following format <copied public key> <space> username@hostname. **NOTES:** ▪ If the authorized_keys file does not exist, create it, for example: **# vi authorized_keys** ▪ If the authorized_keys file exists but does not have write access, change the access, for example: **# chmod 644 authorized_keys**
7.	Change the access on the authorized_keys file back to secure.	**# chmod 600** authorized_keys
Configure Secured TCP Connection		
8.	On the GigaVUE-OS node, configure the secured TCP connection.	**(config) # logging 192.168.1.25 tcp 51300 ssh username sysloguser**
9.	On the GigaVUE-OS	**(config) # show logging**

Step	Description	Command
	node, display the configuration.	

CLI Parameter Limits

This appendix provides information on supported ranges and default values for packet distribution and system parameters in the GigaVUE-OS. Refer to the following sections for details:

- System Parameters
- User Parameters
- CLI limits Second Level Map Parameters
- CLI limits Maximum Nodes per Cluster
- CLI limits GigaStream Maximums
- CLI limits Map Rule Maximums
- Alias Limitations

System Parameters

Parameter	Value
Maximum Characters per line in CLI	8192
Maximum Characters in System Name	64
Maximum Characters in System Prompt Name	64
Remote Timeout (Default)	Default is 15 minutes
Console Baud Rate (Default)	115,200 bps
MTU Size Range (Network/Tool Ports)	The MTU is fixed at 9400 for all network/tool ports on the following platforms: • GigaVUE-HC1 • GigaVUE-HC3 • PRT-HD0-C06X24 line card on GigaVUE-OS HD Series • GigaVUE-TA100, GigaVUE-TA200 and GigaVUE-TA25. **RECOMMENDATION:** Set the MTU to 9400 on all platforms.
Maximum number of TACACS+ Servers per Node	Unlimited*, 5 (recommended)
Maximum number of RADIUS Servers per Node	Unlimited*, 5 (recommended)
Maximum number of SNMP Trap Destinations per	Unlimited*, 5 (recommended)

Parameter	Value
Node	
Maximum number of NTP Servers per Node	Unlimited*, 5 (recommended)
Maximum number of Syslog Servers per Node	Unlimited*, 5 (recommended)
Maximum number of simultaneous sessions to a node by an admin user through any access method (SSH, HTTPS, SCP, SFTP)	Unlimited, based on system resources
Maximum number of simultaneous sessions to a node by a non-admin user through any access method (SSH, HTTPS, SCP, SFTP)	150, based on system resources

NOTE: Unlimited* means that there is no limit, however if you increase beyond the recommended number, system performance may be affected.

User Parameters

User Parameters

Maximum number of user accounts per node	35
user name (minimum and maximum alphanumeric characters)	1 - 30
password	8 - 64

CLI limits Second Level Map Parameters

Second level maps use special **gsrule** and **flow-rules**, as described in GigaSMART Adaptive Packet Filtering (APF) and GigaSMART FlowVUE. A given GigaSMART group can support the following maximums for these special rule types:

- Five gsrules
- Thirty-two flow-rules

CLI limits Maximum Nodes per Cluster

Refer to the *"Cluster Node Limit"* section in the *GigaVUE Fabric Management Guide*.

CLI limits GigaStream Maximums

Refer to the *"Maximum Ports per GigaStream"* section in the *GigaVUE Fabric Management Guide*.

CLI limits Map Rule Maximums

Refer to the *"How Many Map Rules are Supported?"* section in the *GigaVUE Fabric Management Guide*.

Alias Limitations

The maximum number of characters in an alias is 128.

Aliases cannot contain the following special characters: {* " ? ; : , / \ % @ }

Special Character	Description
*	asterisk
@	at sign
\	backslash
/	forward slash
:	colon
,	comma
{	open curly brace
}	closed curly brace
"	double-quote
	space
?	question mark
%	percent sign
;	semi-colon

For example, the following error message is displayed:

(config) port 1/3/x12 alias port\1% Invalid alias 'port\1'. '\' is a reserved character.

NOTE: When backslash (\) and quotation mark (") are used at the very end of an alias, a newline is appended to the alias.

```
(config) # map ali m\>
(config map alias m^J) # from 1/1/g2
```

Additional Sources of Information

This appendix provides additional sources of information. Refer to the following sections for details:

- Documentation
- Documentation Feedback
- Contact Technical Support
- Contact Sales
- The VÜE Community

Documentation

©This table lists all the guides provided for GigaVUE-OS software and hardware. The first row provides an All-Documents Zip file that contains all the guides in the set for the release.

NOTE: In the online documentation, view What's New to access quick links to topics for each of the new features in this Release; view Documentation Downloads to download all PDFs.

Table 1: Documentation Set for Gigamon Products

GigaVUE-OS 6.14 Hardware and Software Guides
DID YOU KNOW? If you keep all PDFs for a release in common folder, you can easily search across the doc set by opening one of the files in Acrobat and choosing **Edit > Advanced Search** from the menu. This opens an interface that allows you to select a directory and search across all PDFs in a folder.
Hardware how to unpack, assemble, rackmount, connect, and initially configure ports the respective GigaVUE-OS devices; reference information and specifications for the respective GigaVUE-OS devices
GigaVUE-HC1 Hardware Installation Guide
GigaVUE-HC3 Hardware Installation Guide
GigaVUE-HC1-Plus Hardware Installation Guide
GigaVUE-HCT Hardware Installation Guide
GigaVUE-TA25 Hardware Installation Guide
GigaVUE-TA25E Hardware Installation Guide
GigaVUE-TA100 Hardware Installation Guide
GigaVUE-TA200 Hardware Installation Guide

GigaVUE-OS 6.14 Hardware and Software Guides
GigaVUE-TA200E Hardware Installation Guide
GigaVUE-TA400 Hardware Installation Guide
GigaVUE-TA400E Hardware Installation Guide
GigaVUE-OS Installation Guide for DELL S4112F-ON
G-TAP A Series 2 Installation Guide
GigaVUE M Series Hardware Installation Guide
GigaVUE-FM Hardware Appliances Guide
Software Installation and Upgrade Guides
GigaVUE-FM Installation, Migration, and Upgrade Guide
GigaVUE-OS Upgrade Guide
GigaVUE V Series Migration Guide
Fabric Management and Administration Guides
GigaVUE Administration Guide covers both GigaVUE-OS and GigaVUE-FM
GigaVUE Fabric Management Guide how to install, deploy, and operate GigaVUE-FM; how to configure GigaSMART operations; covers both GigaVUE-FM and GigaVUE-OS features
GigaVUE Application Intelligence Solutions Guide
GigaVUE Inline Solutions Guide(**NEW**) (previously included in the **GigaVUE Fabric Management Guide**)
Cloud Guides how to configure the GigaVUE Cloud Suite components and set up traffic monitoring sessions for the cloud platforms
GigaVUE V Series Applications Guide
GigaVUE Cloud Suite Deployment Guide - AWS
GigaVUE Cloud Suite Deployment Guide - Azure
GigaVUE Cloud Suite Deployment Guide - OpenStack
GigaVUE Cloud Suite Deployment Guide - Nutanix
GigaVUE Cloud Suite Deployment Guide - VMware (ESXi)
GigaVUE Cloud Suite Deployment Guide - VMware (NSX-T)
GigaVUE Cloud Suite Deployment Guide - Third Party Orchestration

GigaVUE-OS 6.14 Hardware and Software Guides
Universal Cloud TAP - Container Deployment Guide
Gigamon Containerized Broker Deployment Guide
GigaVUE Cloud Suite Deployment Guide - AWS Secret Regions
GigaVUE Cloud Suite Deployment Guide - Azure Secret Regions
Reference Guides
GigaVUE-OS CLI Reference Guide library of GigaVUE-OS CLI (Command Line Interface) commands used to configure and operate GigaVUE HC Series and GigaVUE TA Series devices
GigaVUE-OS Security Hardening Guide
GigaVUE Firewall and Security Guide
GigaVUE Licensing Guide
GigaVUE-OS Cabling Quick Reference Guide guidelines for the different types of cables used to connect Gigamon devices
GigaVUE-OS Compatibility and Interoperability Matrix compatibility information and interoperability requirements for Gigamon devices
GigaVUE-FM REST API Reference in GigaVUE-FM User's Guide samples uses of the GigaVUE-FM Application Program Interfaces (APIs)
Factory Reset Guidelines for GigaVUE-FM and GigaVUE-OS Devices Sanitization guidelines for GigaVUE Fabric Management Guide and GigavUE-OS devices.
Release Notes
GigaVUE-OS, GigaVUE-FM, GigaVUE-VM, G-TAP A Series, and GigaVUE Cloud Suite Release Notes new features, resolved issues, and known issues in this release ; important notes regarding installing and upgrading to this release **Note**: Release Notes are not included in the online documentation. **Note**: Registered Customers can log in to My Gigamon to download the Software and Release Notes from the Software and Docs page on to My Gigamon. Refer to How to Download Software and Release Notes from My Gigamon.
In-Product Help
GigaVUE-FM Online Help how to install, deploy, and operate GigaVUE-FM.

How to Download Software and Release Notes from My Gigamon

Registered Customers can download software and corresponding Release Notes documents from the **Software & Release Notes** page on to My Gigamon. Use the My Gigamon Software & Docs page to download:

- Gigamon Software installation and upgrade images,
- Release Notes for Gigamon Software, or
- Older versions of PDFs (pre-v5.7).

To download release-specific software, release notes, or older PDFs:

1. Log in to My Gigamon.
2. Click on the **Software & Release Notes** link.
3. Use the **Product** and **Release** filters to find documentation for the current release. For example, select Product: "GigaVUE-FM" and Release: "5.6," enter "pdf" in the search box, and then click **GO** to view all PDF documentation for GigaVUE-FM 5.6.xx.

NOTE: My Gigamon is available to registered customers only. Newer documentation PDFs, with the exception of release notes, are all available through the publicly available online documentation.

Documentation Feedback

We are continuously improving our documentation to make it more accessible while maintaining accuracy and ease of use. Your feedback helps us to improve. To provide feedback and report issues in our documentation, send an email to: documentationfeedback@gigamon.com

Please provide the following information in the email to help us identify and resolve the issue. Copy and paste this form into your email, complete it as able, and send. We will respond as soon as possible.

Documentation Feedback Form		
About You	**Your Name**	
	Your Role	
	Your Company	

For Online Topics	**Online doc link**	*(URL for where the issue is)*
	Topic Heading	*(if it's a long topic, please provide the heading of the section where the issue is)*
For PDF Topics	**Document Title**	*(shown on the cover page or in page header)*
	Product Version	*(shown on the cover page)*
	Document Version	*(shown on the cover page)*
	Chapter Heading	*(shown in footer)*
	PDF page #	*(shown in footer)*
How can we improve?	**Describe the issue**	*Describe the error or issue in the documentation.* *(If it helps, attach an image to show the issue.)*
	How can we improve the content? **Be as specific as possible.**	
	Any other comments?	

Contact Technical Support

For information about Technical Support: Go to **Settings** **> Support > Contact Support** in GigaVUE-FM.

You can also refer to https://www.gigamon.com/support-and-services/contact-support for Technical Support hours and contact information.

Email Technical Support at support@gigamon.com.

Contact Sales

Use the following information to contact Gigamon channel partner or Gigamon sales representatives.

Telephone: +1.408.831.4025

Sales: inside.sales@gigamon.com

Partners: www.gigamon.com/partners.html

Premium Support

Email Gigamon at inside.sales@gigamon.com for information on purchasing 24x7 Premium Support. Premium Support entitles you to round-the-clock phone support with a dedicated Support Engineer every day of the week.

The VÜE Community

The VÜE Community is a technical site where Gigamon users, partners, security and network professionals and Gigamon employees come together to share knowledge and expertise, ask questions, build their network and learn about best practices for Gigamon products.

Visit the VÜE Community site to:

- Find knowledge base articles and documentation
- Ask and answer questions and learn best practices from other members.
- Join special-interest groups to have focused collaboration around a technology, use-case, vertical market or beta release
- Take online learning lessons and tutorials to broaden your knowledge of Gigamon products.
- Open support tickets (Customers only)
- Download the latest product updates and documentation (Customers only)

The VÜE Community is a great way to get answers fast, learn from experts and collaborate directly with other members around your areas of interest.

Register today at community.gigamon.com

Questions? Contact our Community team at community@gigamon.com.

Glossary

D

decrypt list

need to decrypt (formerly blacklist)

decryptlist

need to decrypt - CLI Command (formerly blacklist)

drop list

selective forwarding - drop (formerly blacklist)

F

forward list

selective forwarding - forward (formerly whitelist)

L

leader

leader in clustering node relationship (formerly master)

M

member node

follower in clustering node relationship (formerly slave or non-master)

N

no-decrypt list

no need to decrypt (formerly whitelist)

nodecryptlist

no need to decrypt- CLI Command (formerly whitelist)

P

primary source

root timing; transmits sync info to clocks in its network segment (formerly grandmaster)

R

receiver

follower in a bidirectional clock relationship (formerly slave)

S

source

leader in a bidirectional clock relationship (formerly master)