Gigamon Containerized Broker Guide

GigaVUE Cloud Suite

Product Version: 5.15

Document Version: 1.0

(See Change Notes for document updates.)

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
5.15.00	1.0	03/30/2022	Original release of this document with 5.15.00 GA.

Contents

Gigamon Containerized Broker

Gigamon Containerized Broker (GCB) is a containerized component that provides the network broker features in a containerized form. GCB can perform traffic acquisition, aggregation, basic filtering, replication, and tunneling with encryption support. GCB can be deployed in its own POD as a Kubernetes service where your workloads are running. There are various components based on multiple scenarios and requirements that the GCB receives the traffic from.

This guide provides an overview of Gigamon Containerized Broker and describes how to install and deploy GCB components in your PODs.

Topics:

About Gigamon Containerized Broker

The Gigamon Containerized Broker (GCB) is a containerized component that provides the network broker features in a containerized form. The GCB is deployed by Kubernetes orchestrator and not by GigaVUE-FM.

GCB initiates the traffic acquisition process with GCB PODs and enhances the support of the features.

Following are the modules implemented in GCB:

- **Traffic Acquisition using CNI Modules**: GCB supports traffic acquisition by reading the traffic from the Container Network Interface (CNI) modules like AWS ENI, Calico, and Flannel. During initialization, GCB receives the configuration information from the Gigamon's YAML file. Kubernetes CNI (Container Network Interface) supports any combination of ingress, egress, and management process. Following the specifications defined in the YAML file, GCB configures itself on your worker node to acquire traffic.

  NOTE: After GCB registration, you cannot change the number of CNI, and CNI types. If required, a new GCB instance configured and registered.

- **Traffic Aggregation** - When GCB is running in its own POD, GCB itself serves as a traffic aggregator.
- **Filtering Module** - GCB allows basic filtering, forwarding policy, and enrichment. GCB's filtering can be passed from the YAML file, and it is based upon the protocol. The filters and rules are pushed to GCB from GigaVUE-FM and can be modified while the GCB is running.
- **Tunneling Modules** - GCB supports L2GRE and VXLAN tunneling modules.
- **Encryption Module** - GCB maintains the required certificates to support TLS and HTTPS encryption.

GCB and GigaVUE-FM Interaction

Following are the interactions between GCB and GigaVUE-FM:

- GCB Registration
- GCB Deregistration
- GCB Heartbeats
- GCB Statistics
- Monitoring Domain and Traffic Policy

GCB Registration

When GCB comes up in the Kubernetes environment, GCB registers itself with GigaVUE-FM. When GigaVUE-FM is unreachable, GCB tries to connect with five retries of increasing time periods. If the GigaVUE-FM is unreachable even after the retries, Kubernetes deployment of GCB fails. GCB only supports IPv4 protocol.

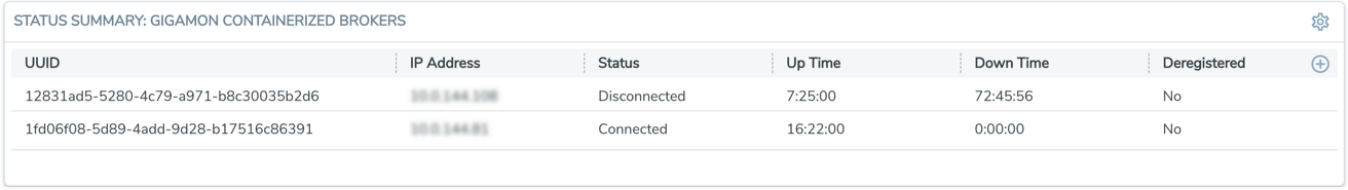
GCB Deregistration

When GCB is terminated normally, GCB sends the deregistration message to GigaVUE-FM. If GCB goes down abnormally, it might not get deregistered. The GCB PODs associated to a GCB node might then get moved to the other GCB node. Similarly, if a GCB goes down, the feeding G-vTAPs are moved to the other GCB, and the GigaVUE-FM does not store information of the GCB POD.

GCB Heartbeats

Periodically, GCB sends heartbeats to GigaVUE-FM. By default, the status of GCB is marked as **Connected**. The following are the various scenarios where the GCB status changes:

- If 3 consecutive heartbeats are missed, GigaVUE-FM marks the status as **Disconnected**.
- If 2 consecutive heartbeats are missed, GigaVUE-FM marks the status as **Pending**.
- If GigaVUE-FM does not receive GCB heartbeats for 30 days, then GigaVUE-FM removes the GCB, considering it as stale.


STATUS SUMMARY: GIGAMON CONTAINERIZED BROKERS

UUID	IP Address	Status	Up Time	Down Time	Deregistered
12831ad5-5280-4c79-a971-b8c30035b2d6	[redacted]	Disconnected	7:25:00	72:45:56	No
1fd06f08-5d89-4add-9d28-b17516c86391	[redacted]	Connected	16:22:00	0:00:00	No

GCB Statistics

GCB sends traffic statistics and associated GCB PODs to GigaVUE-FM. The highest traffic and lowest traffic widgets in GigaVUE-FM dashboard shows the details of 10 highest and 10 lowest GCB traffic statistics.

GCB continues to send the statistics even when there is no traffic flowing. The GCB statistics are not stored in cache even when GigaVUE-FM is not reachable by GCB at that instant of time.

Monitoring Domain and Traffic Policy

You can configure and manage the Monitoring Domains, Traffic Policies, Connections, Metadata fields, and Source Inventories of GCB in GigaVUE-FM. Refer to the *GigaVUE-FM REST API Reference* for detailed information on the REST APIs of GCB.

- A Traffic Policy is a combination of Rules and Tunnels.
- A rule contains specific filtering criteria that the packets must match. The filtering criteria lets you determine the target instances and the (egress or ingress) direction of tapping the network traffic.
- A tunnel is a communication path in which the traffic matching the filtered criteria is routed to the destination.

GCB for Service Mesh and HTTPS/2 Support with Metadata

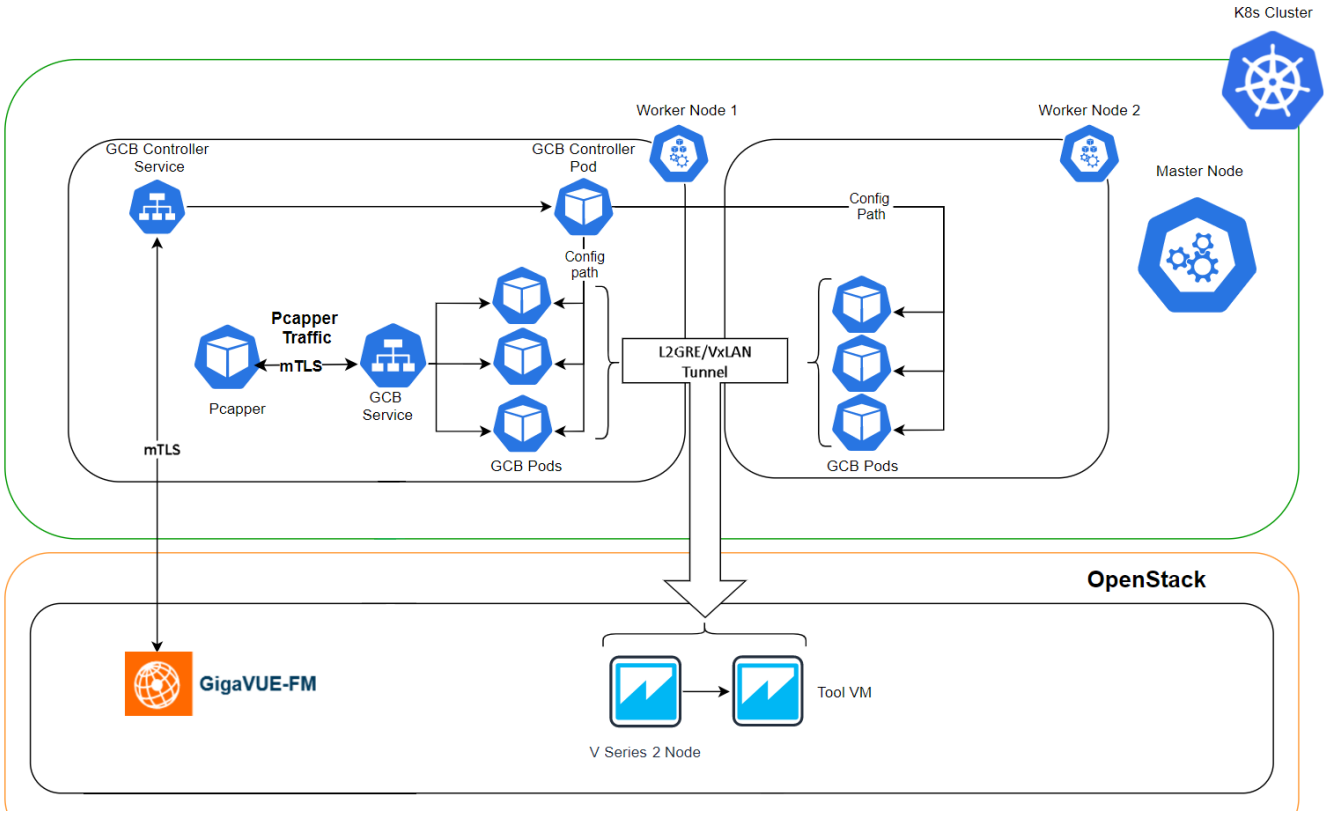
This guide provides an overview of Gigamon Containerized Broker for service mesh and HTTPS/2 support with metadata and describes how to install and deploy GCB components.

Refer to the following topics for details:

- Architecture of GCB for Service Mesh and HTTPS/2 Support with Metadata
- Get Started with GCB for Service Mesh and HTTPS/2 Support with Metadata
- Configure GCB for Service Mesh and HTTPS/2 Support with Metadata

Architecture of GCB for Service Mesh and HTTPS/2 Support with Metadata

The following diagram illustrates the architecture of Gigamon Containerized Broker for service mesh and HTTPS/2 support with metadata environment.



1. The GCB Controller is registered with GigaVUE-FM and the traffic policy is deployed on the GCBs.
2. Communication of configuration, data, and statistics to and from GCB is performed through the GCB Controller Service. GigaVUE-FM communicates with the GCB PODs through the GCB Controller.
3. Each GCB POD is registered with GigaVUE-FM and the traffic policy is deployed on the GCBs.
4. The Pcapper collects the network traffic and sends the HTTP packets to GCB service through mTLS authentication. Refer to Prerequisites for mTLS authentication for detailed information.
5. In the GCB service, the received HTTP packets are load balanced across the available GCB PODs.
6. GCB PODs filters the packets based on the metadata.
7. The filtered HTTP packets from GCB PODs are tunneled directly to the Tools or through the V Series nodes on OpenStack environment. Refer to the *GigaVUE Cloud Suite for OpenStack Configuration Guide* for more information on V Series configuration on OpenStack environment.
8. GCB Controller collects the data from GCB PODs and sends the collected statistics and heartbeats to GigaVUE-FM through mTLS authentication. Refer to Prerequisites for mTLS authentication for detailed information.

Get Started with GCB for Service Mesh and HTTPS/2 Support with Metadata

This section describes how to initiate GCB and GigaVUE-FM deployment with the required licenses and network requisites.

Refer to the following sections for details:

- Components of GCB for Service Mesh and HTTPS/2 Support with Metadata
- License Information
- Network Requirements
- Configure mTLS authentication

Components of GCB for Service Mesh and HTTPS/2 Support with Metadata

The Gigamon Containerized Broker for service mesh and HTTPS/2 support with metadata works with the following components:

- **GigaVUE® Fabric Manager (GigaVUE-FM)** is a web-based fabric management and orchestration interface that provides a single pane of glass visibility, management, and orchestration of both the physical and virtual traffic that form the GCB.
- **GCB Service** is a traffic acquisition component that collects the data from the Pcapper and sends the Pcapper traffic to the GCB PODs.
- **GCB POD** is the primary GCB module that collects the Pcapper traffic from GCB Service, filters the traffic and tunnels the filtered traffic directly to the tools or through the V Series nodes. GCB POD also sends the statistics and heartbeats to GCB Controller.
- **GCB Controller** is the management component of GCB to control and communicate with GCB PODs. GCB Controller collects the data from GCB PODs and sends the collected statistics and heartbeats to GigaVUE-FM.

License Information

All the GCB PODs deployed in your environment periodically report the statistics to GCB Controller. Then the GCB Controller periodically reports the collective statistics of GCB PODs to GigaVUE-FM for Volume-Based Licensing. GigaVUE-FM adds the required licensing tags into the Elasticsearch.

In the Volume-Based Licensing scheme, a license entitles specific applications on your devices to use a specified amount of total data volume over the term of the license. The distribution of the license to individual nodes or devices becomes irrelevant for Gigamon's

accounting purpose. GigaVUE-FM tracks the total amount of data processed by the various licensed applications and provides visibility into the actual amount of data, each licensed application is using on each POD, and tracks the overuse if any.

Network Requirements

The following table describes the Kubernetes network requirements for GCB to work efficiently.

Direction	Type	Protocol	Port	CIDR	Purpose
Gigamon Containerized Broker deployed inside Kubernetes worker node					
Outbound	HTTPS	TCP	443	Any IP address	Allows GCB Controller to communicate with GigaVUE-FM
Inbound	HTTPS	TCP	8443 (configurable)	Any IP address	Allows GigaVUE-FM to communicate with GCB Controller.
Outbound	HTTPS	TCP	42042	Any IP address	Allows GigaVUE-FM to communicate with GCB to send statistics data.

Configure GCB for Service Mesh and HTTPS/2 Support with Metadata

Setting up GCB for Service Mesh and HTTPS/2 Support with Metadata involves the following two steps:

- Implement GCB in Kubernetes
- Configure GCB for Service Mesh and HTTPS/2 Support with Metadata through GigaVUE-FM

> The Red Hat supported base images of the GCB applications are built on the top of Red Hat Universal Base Image or Red Hat Enterprise Linux Image. The GCB images are **Red Hat Certified** for Red Hat OpenShift platform.

Implement GCB in Kubernetes

To fully implement GCB, the following eight steps are required to be completed:

1. Implement external access to the Kubernetes environment (e.g., ingress, external public IPs, load balancers) to allow communication between GCB and GigaVUE-FM.

2. Ensure that the firewall rules on Kubernetes nodes are met according to the Network Requirements.
3. Implement mTLS. Refer to Configure mTLS Authentication.
4. Add the GCB images to a private Docker registry or ensure that the files can be pulled from the Docker Hub registry.
5. Deploy GCB Controller Service.
6. Deploy GCB Controller PODs.
7. Deploy GCB HTTP Service.
8. Deploy GCB HTTP PODs.

Deploy GCB Controller Service

Follow the instructions below to deploy GCB Controller Service in your Kubernetes environment:

> **NOTE:** Contact Contact Technical Support or Contact Sales for the GCB images and YAML files.

1. The following is sample data that can be entered into your YAML file. Edit your YAML file based on the sample given below. Do not copy and paste this content into your YAML file:

```
apiVersion: v1
kind: Service
metadata:
  name: gigamon-gcb-cntlr-service
  labels:
    app: gcb-cntlr
    service: gigamon-gcb-cntlr-service
    change the namespace to match your namespace
  namespace: default
spec:
  ports:
  - port: 8443
    protocol: TCP
    name: gcb-rest
    targetPort: 8443
  - port: 42042
    protocol: TCP
    name: gcb-stats
    targetPort: 42042
  selector:
    app: gcb-cntlr
```

   The following table gives a description of all the field values in the YAML file that are updated:

Field Values	Description
Port: 8443	The GCB Controller REST service port number.
Port: 42042	This port must be port 42042. This allows GigaVUE-FM to communicate with GCB to send statistical data.

2. Using the YAML file, Kubernetes creates the GCB Controller Service.

Deploy GCB Controller PODs

Follow the instructions below to deploy GCB Controller Service in your Kubernetes environment:

> **NOTE:** Contact Contact Technical Support or Contact Sales for the GCB images and YAML files.

1. In your Kubernetes orchestrator, edit the GCB Controller image name, commands, and other required information into your YAML file. The following is sample data that can be entered in your YAML file. Edit your YAML file based on the sample given below. Do not copy and paste this content into your YAML file:

```
name: gcb-cntlr
image: gigamon/gcb-cntlr:gcb-cntlr:<version>
command:
- # /gcb-cntlr
- # <FM IP>
- # <FM REST Svc Port>
- # <GCB-Cntlr REST SVC Port>
- # <mTLS Mode: 1(ON)|0(OFF))
- # <Cert Path>
- # <Cert file>
- # <Pvt Key>
- # <CA-Root>
imagePullPolicy: Always
ports:
- containerPort: 8443
- containerPort: 42042
env:
# Service name.Should match name specified in metadata section.
- name: GCB_CNTLR_SERVICE_NAME
 value: "GIGAMON_GCB_CNTLR_SERVICE"
# External LB balancer IP, for controller (FM) to connect to gcb-cntlr
- name: GCB_CNTLR_EXT_IP_DNS
 value: "<external IP for GigaVUE-FM to reach GCB CNTLR>"
# K8S cluster end-point
- name: K8S_CLUSTER_ENDPOINT
 value: "https://<kubernetesapiserverurl>:6443"
# Namespace of pod
- name: GCB_CNTLR_POD_NAMESPACE
 valueFrom:
      fieldRef:
          fieldPath: metadata.namespace
```

The following table gives a description of all the field values in the YAML file that are changed or updated:

Field Values	Description
/gcb-cntlr (image name)	GCB Controller image name and version. Make sure to use the latest image version.
GigaVUE-FM IP	The IP address of the GigaVUE-FM with which your GCB is connected.
FM REST Svc Port	The FM REST service port number. This must be opened on your Kubernetes to allow outbound traffic. This allows GCB Controller to communicate with GigaVUE-FM. Example: 443
GCB-Cntlr REST SVC Port	The GCB Controller REST service port number. This must be opened on your GigaVUE-FM to allow inbound traffic to Kubernetes. This allows GigaVUE-FM to communicate with GCB Controller.

Field Values	Description
	Example: 8443
mTLS Mode: 1(ON) | 0(OFF)	To specify if mTLS mode between GigaVUE-FM and GCB controller should be On or Off. Values are: • 1 - ON • 0 - OFF
Cert Path	Path of the certificate file. Example: /etc/gcbcerts
Cert file	Name of the certificate file. Example: gcb-cert.pem
Pvt Key	Name of the private key. Example: gcb-pvt-key.pem
CA-Root	Name of the CA root certificate. Example: gcb-ca-root-cert.pem
Ports: o containerPort: 8443 o containerPort: 42042	Two ports must be opened. The first container port must be the same as GCB-Cntlr REST SVC Port. The second container port must be port 42042. This allows GigaVUE-FM to communicate with GCB to send statistics data.
External LB balancer IP	The external load balancer IP/DNS value to allow GigaVUE-FM to communication with GCB Controller within Kubernetes. The GigaVUE-FM IP entry may change when you upgrade or redeploy.
K8S cluster end-point	Kubernetes cluster end point for GigaVUE-FM to access the control plane. Example: https://<kubernetesapiserverurl>:6443

2. Using the YAML file, Kubernetes automatically downloads the defined GCB Controller PODs and deploys it to the Kubernetes worker node.

Deploy GCB HTTP Service

Follow the below instructions to deploy GCB HTTP service in your Kubernetes environment:

> **NOTE:** Contact Contact Technical Support or Contact Sales for the GCB images and YAML files.

1. The following is sample data that can be entered into your YAML file. Edit your YAML file based on the sample given below. Do not copy and paste this content in your YAML file:

```
apiVersion: v1
kind: Service
metadata:
  name: gcb-http-service
  labels:
    app: gcb-http
    service: gcb-http-service
  # change the namespace to match your namespace
  namespace: default
spec:
  ports:
  - port: 9443
    name: https
  selector:
    app: gcb-http
```

   The following table gives a description of all the field values in the YAML file that is updated:

Field Value	Description
9443	The GCB Controller REST service port number. This must be opened on your GigaVUE-FM to allow inbound traffic to Kubernetes.

2. Using the YAML file, Kubernetes creates the defined GCB HTTP service.

Deploy GCB HTTP PODs

Follow the instructions below to deployGCBHTTP PODs in your Kubernetes environment:

> **NOTE:** Contact Contact Technical Support or Contact Sales for the GCB images and YAML files.

1. In your Kubernetes orchestrator, edit the GCBHTTP POD image name, commands, and other required information in a YAML file. The following is sample data that can be entered into your YAML file. Edit your YAML file based on the sample given below.Do notcopy and paste this content into your YAML file:

```
name: gcb-http
command:
- # /gcb-http
- # PORT for RX
- # mTLS-Flag(T/F)
- # CERT_FILE
- # KEY_FILE
- # CA_CERT_FILE
- # CA_VERIFY(T/F)
- # default destination ip (if not configured from GigaVUE-FM)
- # (1=> default, 0=> rule)
- # (1=> L2GRE, 3=> VXLAN)
image: gigamon/gcb-http:<version>
imagePullPolicy: Always
env:
- name: GCB_DEBUG_MODE
value: "0x031A2F14"
- name: GCB_SERVICE_NAME
  value: "GIGAMON_GCB_HTTP2_SERVICE"
- name: GCB_CNTLR_SVC_DNS
  #value: "<GCB-CNTLR-SVC-NAME.GCB-CNTLR-NAMESPACE>.svc.cluster.local"
  value: "gigamon-gcb-cntlr-service.default.svc.cluster.local"
- name: GCB_CNTLR_REST_SVC_PORT
# port used to receive configuration from FM
  value: '8443'
- name: GCB_POD_NAMESPACE
valueFrom:
fieldRef:
fieldPath: metadata.namespace
```

The following table gives a description of all the field values in the YAML file that are changed or updated:

Field Value	Description
PORT for RX	HTTP port number for ingress traffic Example: 9443
mTLS-Flag (True/False)	Enable or disable mTLS between Pcapper and GCB.
CERT_FILE	SSL/TLS certificates Example: server-certificate-chain.pem
KEY_FILE	Private key for the certificate Example: server-private-key.pem
CA_CERT_FILE	CA root certificate Example: ca-root-crt-chain.crt

Field Value	Description
CA_VERIFY (True/False)	Enable or disable verification of the certificate files.
default destination ip	Default Destination IP (if not being configured from FM)
(1=> default, 0=> rule)	(0/1) Enter 1 to use the default destination IP, or enter 0 to use the rules configured by GigaVUE-FM
(1=> L2GRE, 3=> VXLAN)	(1/3) Enter 1 to use the L2GRE tunnel type, or enter 3 to use the VXLAN tunnel type.
gigamon/gcb-http:<version>	GCB Controller image name and version. Make sure to use the latest image version.

GCB_DEBUG_MODE	The hex value for GCB debugging. This value must be in the **0xdd[aaaa][b][c]** format, where:
	• aaaa is a hex value for the number of pcap messages to maintain before rollover
	• b is 0 = do not create pcap or 1 = create pcap
	• c is level. Level with 1 =fatal, 2 =error, 3 =info, 4 =debug
	• dd is the log file size multiplier
	  • dd = 0|1 - means default log file size (approx. 100,000 lines)
	  • dd = 08 - means 8 * default log file size (approx. 8*100,0000 lines)
	  • dd = FF = 255 - means (255*100,000 lines)

GCB_CNTLR_SVC_DNS	GCB Controller Service Number. This value must match the metadata used for GCB Controller. Example: gigamon-gcb-cntlr-service.default.svc.cluster.local
GCB_CNTLR_REST_SVC_PORT	The GCB Controller REST service port number. This must be opened on your GigaVUE-FM to allow inbound traffic to Kubernetes.

2. Using the YAML file, Kubernetes automatically downloads and deploys the defined GCB HTTP PODs.

Configure GCB for Service Mesh and HTTPS/2 Support with Metadata through GigaVUE-FM

This section describes how to configure GCB through GigaVUE-FM GUI. Refer to the following section for details.

- Launch GigaVUE-FM
- Create Metadata Field Names
- Create Monitoring Domain

- Configure Service Identification
- Configure Traffic Policy

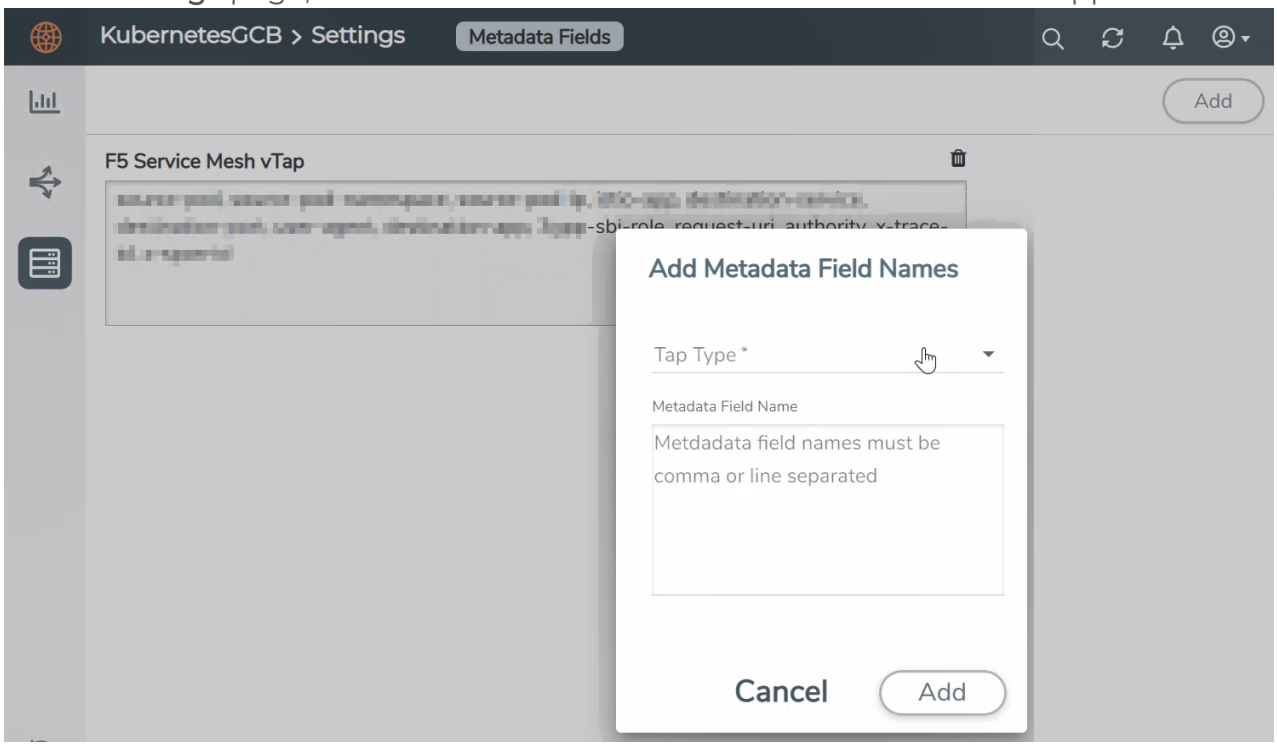
Launch GigaVUE-FM

The recent GigaVUE-FM image files can be downloaded from Gigamon Customer Portal. After fetching the image, upload and launch GigaVUE-FM on your OpenStack environment. For assistance, Contact Technical Support of Gigamon or refer to the *GigaVUE Cloud Suite for OpenStack Guide* for more information on V Series configuration on OpenStack environment.

Create Metadata Field Names

To create metadata field names in GigaVUE-FM:

1. In GigaVUE-FM, on the left navigation pane, select **Inventory > VIRTUAL > KubernetesGCB > Settings**. The **Settings** page appears.
2. In the **Settings** page, click **Add**. The **Add Metadata Field Names** wizard appears.

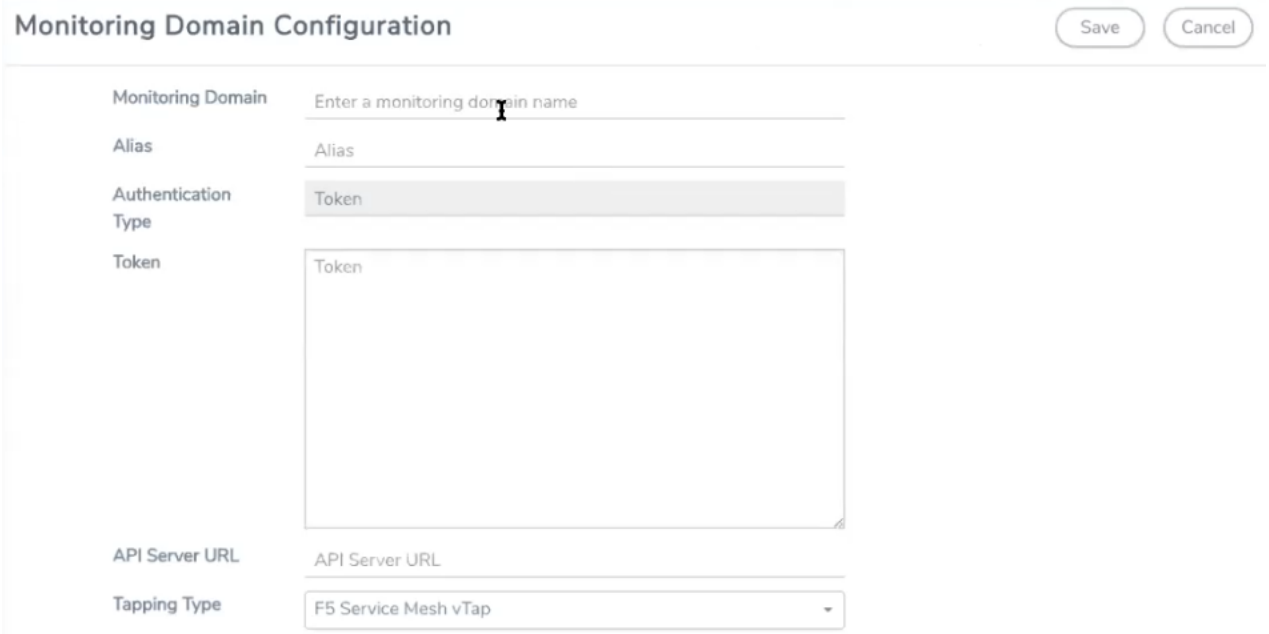


3. Select the **Tap type** as **F5 Service Mesh vTAP** and enter the **Metadata Field Names**.
4. Click **Add**. The newly added metadata field names appear on the **Settings** page.

Create Monitoring Domain

To create a monitoring domain in GigaVUE-FM:

1. In GigaVUE-FM, on the left navigation pane, select **Inventory > VIRTUAL > KubernetesGCB > Monitoring Domain**. The **Monitoring Domain** page appears.
2. In the Monitoring Domain page, click **New**. The Monitoring Domain Configuration wizard appears.



3. Enter or select the required information as described in the following table,

Fields	Description
Monitoring Domain	Enter a name for the monitoring domain
Alias	Enter a name for the GCB connection
Authentication Type	Select Token as the authentication type
API Server URL	Enter the URL of the API server
Tapping Type	Select **F5 Service Mesh vTap** as the Tapping Type

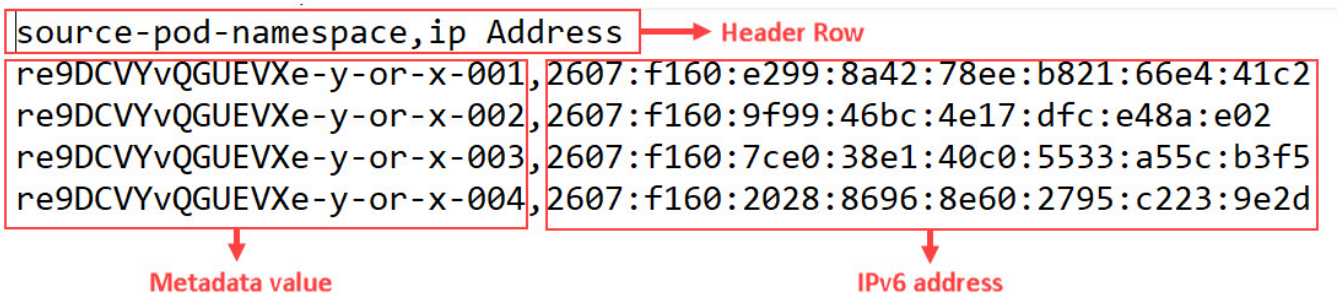
4. Click **Save** to create a monitoring domain.

Configure Service Identification

In the Service mesh and HTTP/s supported platform, the GCB receives packets to be monitored in the form of HTTPS/2 requests. On receiving the HTTPS/2 request from Pcapper, GCB applies the rules configured in GigaVUE-FM and forwards the filtered traffic to V Series 2.x nodes deployed on the OpenStack platform through L2GRE or VXLAN tunnels.

In a Kubernetes environment, the IP addresses associated with pods and services are temporary and can change regularly. For the external tools, these changing IP addresses are difficult to consistently correlate incoming data to the services and the sources related to that data. The same IP addresses may also exist in multiple Kubernetes clusters adding difficulty in identifying the true source of the monitored traffic. To correlate these temporary and same IP addresses, the GigaVUE-FM and GCB use information supplied in the .csv text files to map the temporary IP addresses to IPv6 addresses that the external tools can consistently use.

The CSV file must contain a header row with two columns. The first column is for the Metadata value and the second column is for the IPv6 address. The metadata value specified in the header row and the values in the first column of the CSV file must match theMetadata Field Names.



The length of the metadata value in first column of the non-header row must be less than or equal to 127 and the number of non-header entries (rows) must be less than 4096.

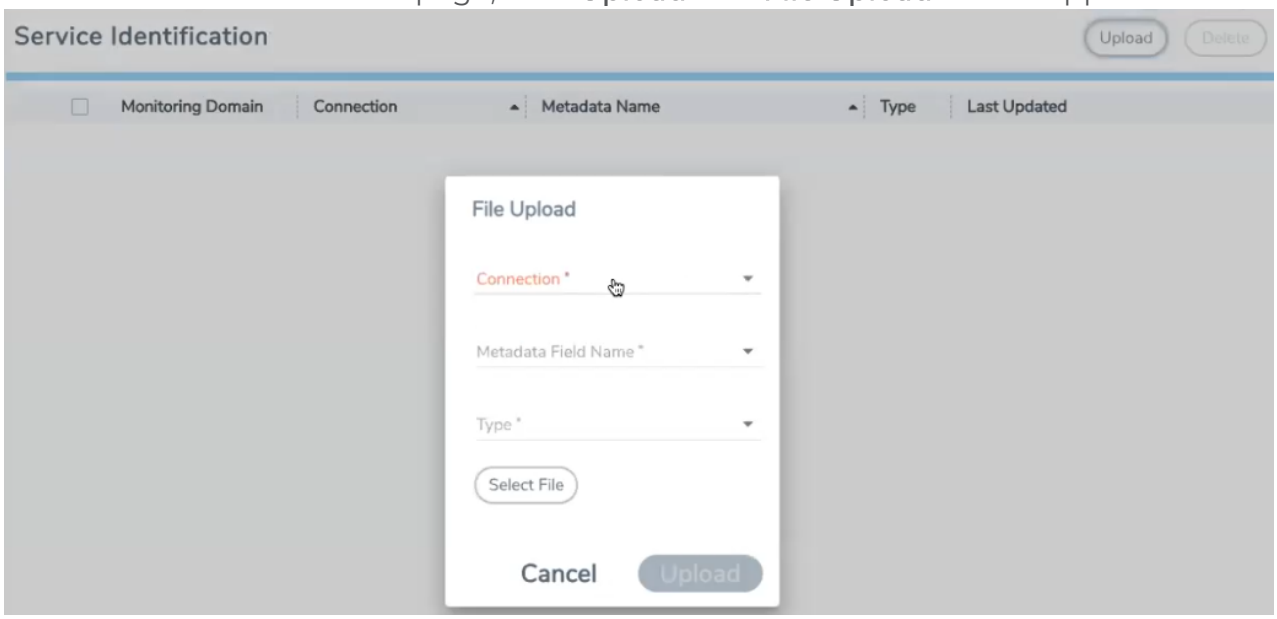
The Service Mesh and HTTPS/2 supported platform must provide the following CSV files:

- a **src-ip-mapping.csv** file to translate the temporary source IP (IPv4) address to an external IPv6 address.
- a **dest-ip-mapping.csv** file to translate the temporary destination IP (IPv4) address to an external IPv6 address.

To upload the mapping CSV files to GigaVUE-FM:

1. After creating a Monitoring Domain, in GigaVUE-FM, from the left navigation pane, select **Inventory > VIRTUAL > KubernetesGCB > Service Identification**. The **Service Identification** page appears.

2. In the Service Identification page, click **Upload**. The **File Upload** wizard appears.



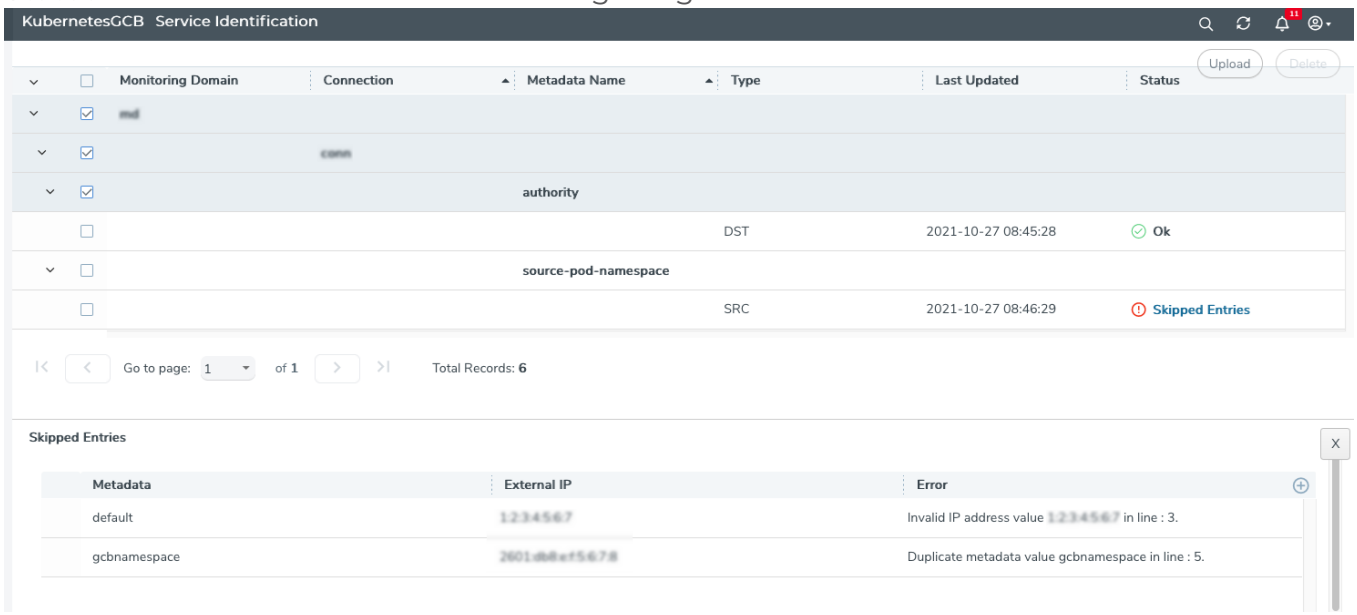
3. Enter or select the required information as described in the following table:

Fields	Description
Connection	Select an existing monitoring domain
Metadata Field Name	Select a Metadata field to search in the CSV file. • If the value for the metadata field matches the content of the received packets, then GCB use the mapping tables to convert the ephemeral IPv4 addresses to external IPv6 addresses and replaces the incoming IPv4 header with an IPv6 header, before forwarding the packets to the Tools or V Series nodes. • If the value for the metadata field doesn't match the content of the received packets, then the GCB forwards the packets without translation.
Type	Select an IP address type from the following: • SRC - Source IP • DST - Destination IP
Select (CSV) File	Select an IP mapping CSV file to upload to GigaVUE-FM.

4. Click **Upload** to upload the selected CSV file for the monitoring domain.

> **NOTE:** You must upload a source and a destination IP mapping CSV file for the IP translation.

Once the CSV file is uploaded successfully, GigaVUE-FM displays the status of the uploaded file. If no error is found in the meta-data, then the status is displayed as **Ok**. However, if there is any error in the meta data or processing, then the error message is displayed under the **Status** column. Click on the error message to get detailed information about the error.



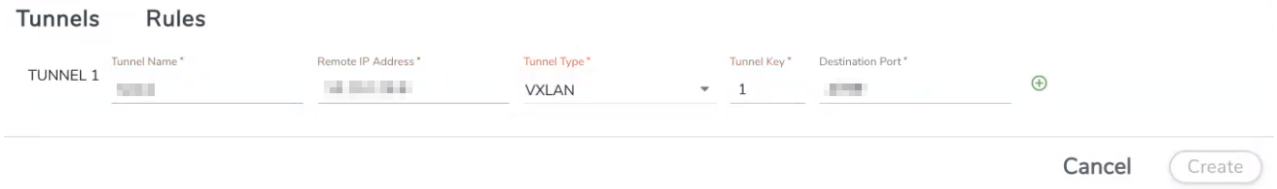
Types of Error messages:

1. **Skipped Entries**: This error message is displayed:
   - If the metadata value is blank or more than 127 characters.
   - If the IP address is invalid.
   - If there are more than 4096 entries in the file excluding the header. In this case, only the first 4096 entries will be sent to GCB and the rest would be skipped.
   - If the uploaded CSV file contains two or more identical entries, or two or more entries with the same meta data values. In this case only the first entry will be sent to GCB and the rest would be skipped.
2. **GCB**: This error message is displayed due to processing errors or a failure.

Configure Traffic Policy

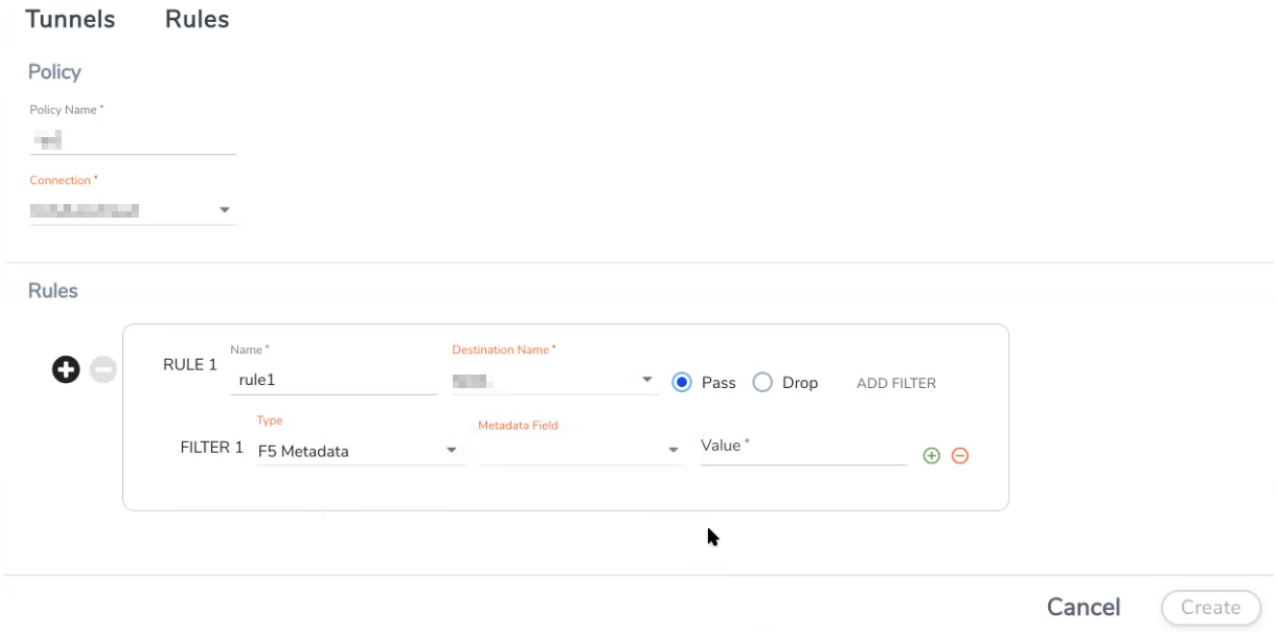
To create a Traffic Policy in GigaVUE-FM:

1. From the GigaVUE-FM left navigation pane, select **Traffic > VIRTUAL > Orchestrated Flows > KubernetesGCB**. The **Orchestrate** page appears with the **Traffic Policy** tab.
2. In the Traffic Policy tab, click **Create**. The Create Tunnels and Rules wizard appears.
3. In the **Tunnels** tab, enter or select the required information as described in the following table:



Fields	Description
Tunnel Name	Enter a name for the Tunnel.
Remote IP Address	Enter an IP Address for the Tunnel.
Tunnel Type	Select L2GRE or VXLAN as the tunnel type.
Tunnel Key	Enter a value for the tunnel key.
Destination Port	If the tunnel type is VXLAN, enter the tunnel destination port number.

4. Switch to **Rules** tab, and enter or select the required information as described in the following table:



Fields	Description
Policy	
Policy Name	Enter a name for the policy.
Connection	Select a connection for the policy.
Rules	
Name	Enter a name for the Rule.
Destination Name	Select a tunnel destination.
Pass/Drop	Select Pass to allow the packets or select Drop to block the packets based on the filters.
Click **ADD FILTER** to add filters for the rule.	
Type	Select the type as F5 Metadata.
Metadata Field	Select a Metadata field name.
Filter value	Enter a value for the filter type.

5. Click **Create** and this new Traffic Policy deploys itself in the GCB.

The Traffic Policy processes the Pcapper traffic and forwards the traffic to the tunnel destination IP address.

View GCB Specifications in GigaVUE-FM

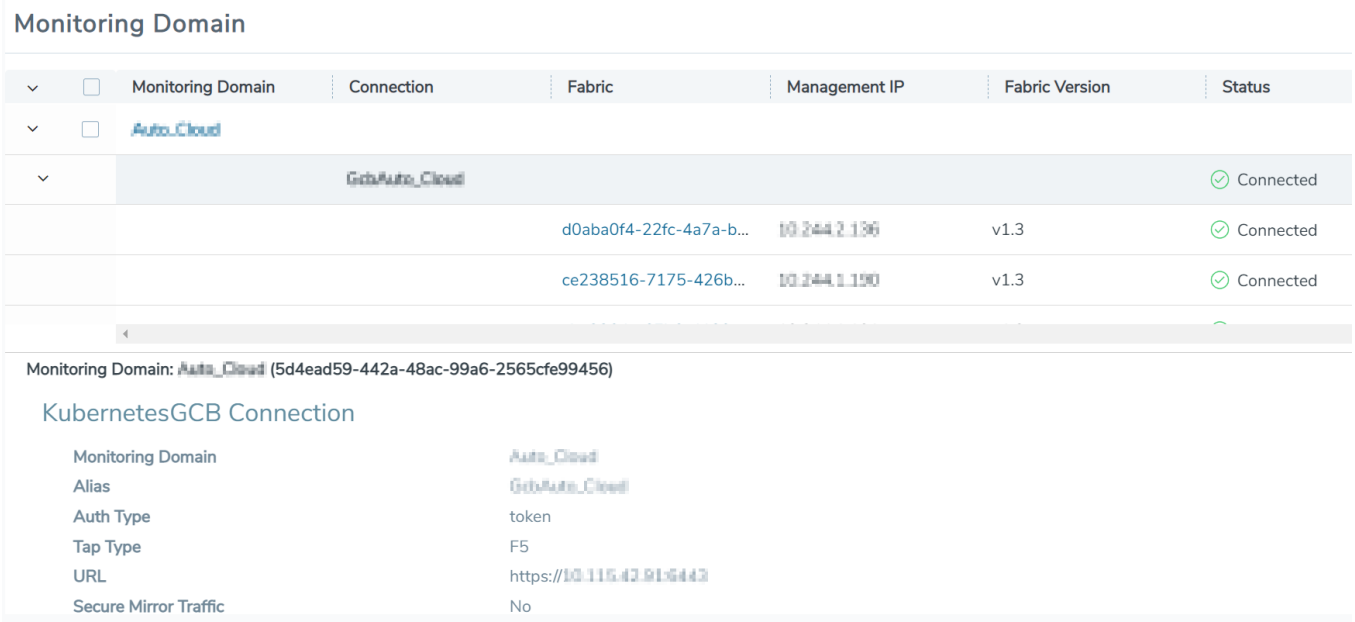
After the GCB configuration, GCB periodically sends the statistics to GigaVUE-FM. In the GigaVUE-FM, you can view the list of available Monitoring Domains, Source Inventories, and Traffic Policies. Refer to the following topics for detailed information.

- View GCB Monitoring Domain
- View Source Inventory
- View GCB Specifications in GigaVUE-FM
- View GCB Log Level Settings

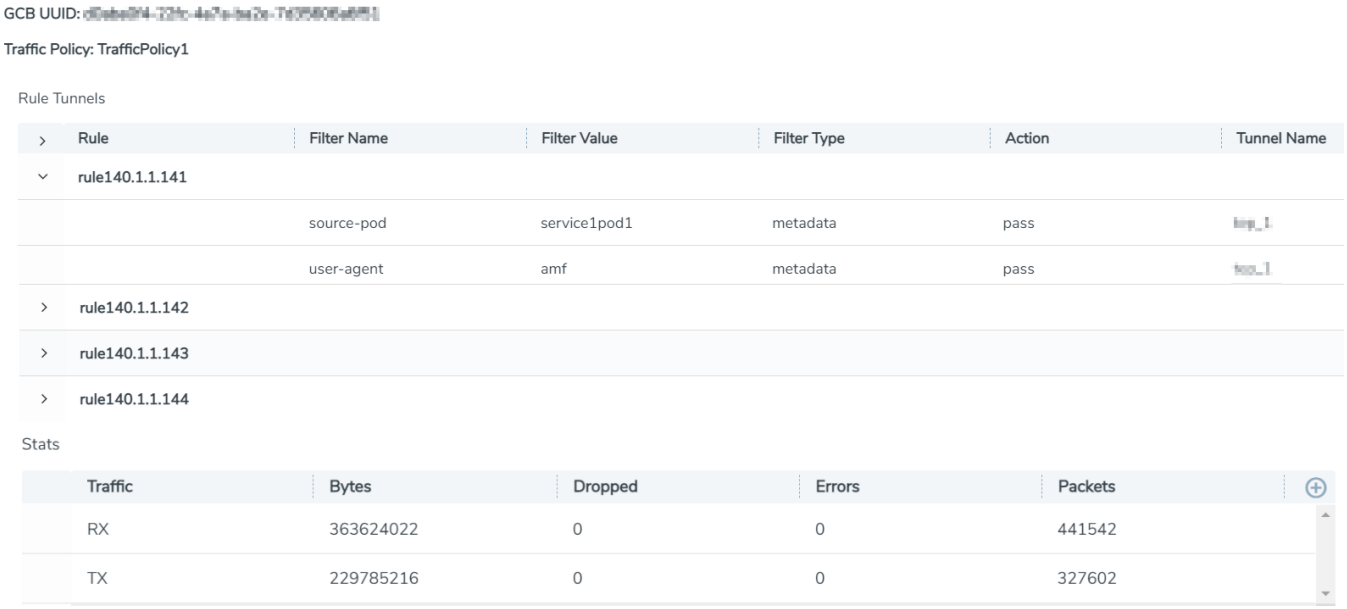
View GCB Monitoring Domain

To view the Monitoring Domains of GCB in GigaVUE-FM, navigate to **Inventory > VIRTUAL > KubernetesGCB > Monitoring Domain**. The Monitoring Domain page appear with the list of Monitoring Domains.

In the Monitoring Domains list, click on a Monitoring Domain name to view the details of the selected Monitoring Domain.

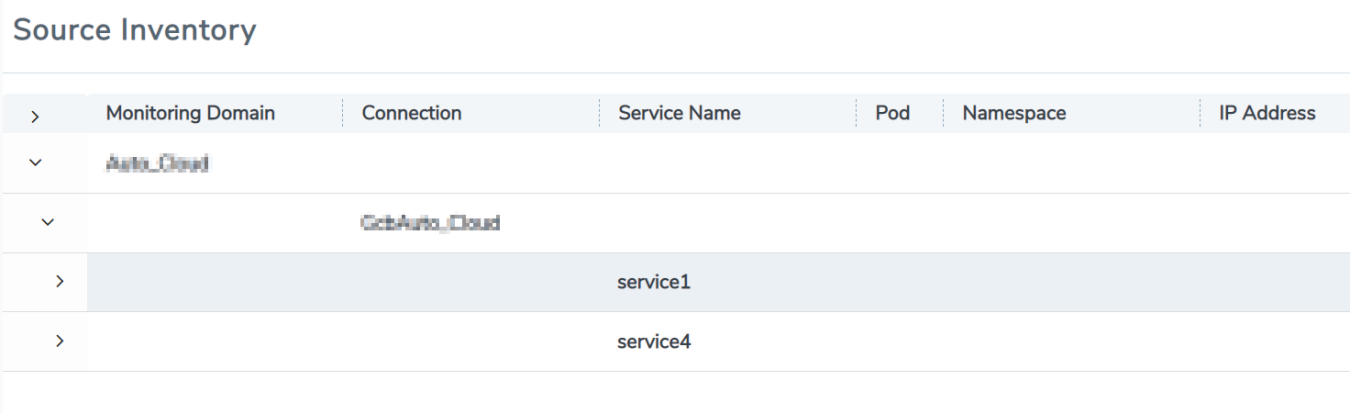


From any existing Monitoring Domain cluster, click on a GCB fabric to view the Rule Tunnels and statistics.



View Source Inventory

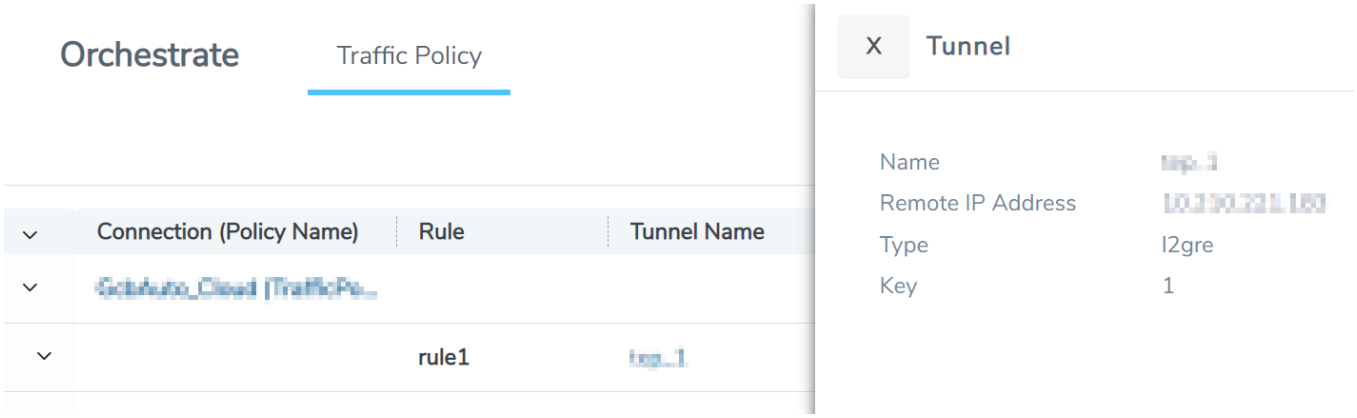
To view the Kubernetes Cluster Source Inventory of GCB in GigaVUE-FM, navigate to **Inventory > VIRTUAL > KubernetesGCB > Source Inventory**. The Source Inventory page appears with the list of Kubernetes cluster inventories. You can add, edit, or delete the Source Inventory of Kubernetes clusters only through REST APIs but not through GigaVUE-FM GUI.



View GCB Traffic Policy

To view the Traffic Policies (Monitoring Sessions) of GCB in GigaVUE-FM, navigate to **Traffic > VIRTUAL > Orchestrated Flows > KubernetesGCB**. The Traffic Policy page appears with the list of Traffic Policies.

From any existing Traffic Policy, click on the Tunnel Name. The Tunnel quick view appears with the details of the selected tunnel.



Click on the Connection (Policy Name) to view the status of the last operation performed on the policy.

View GCB Log Level Settings

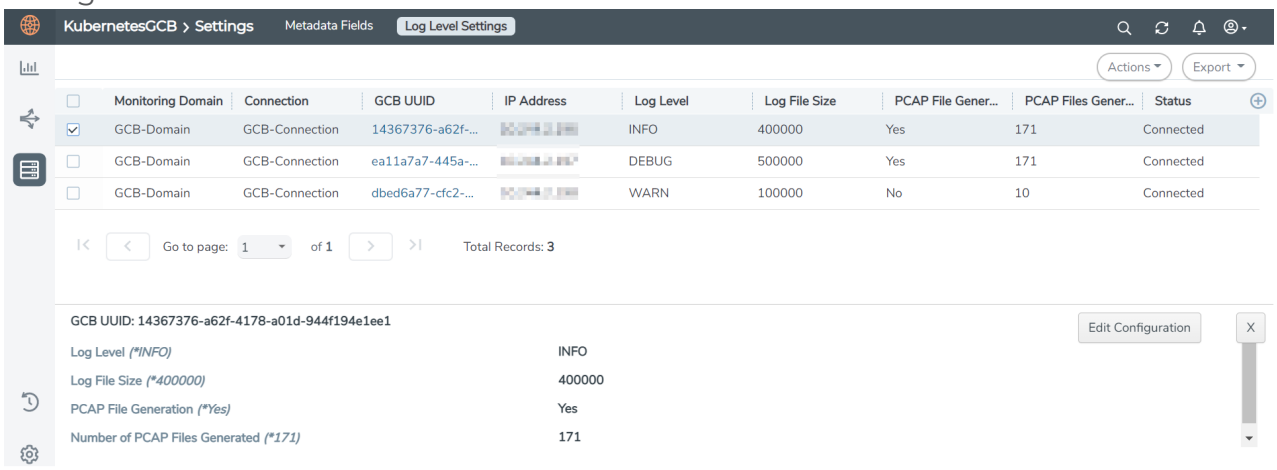
> **NOTE:** Early Access is a pre-GA feature status indicating it is not recommended for production networks. The functionality has been tested. Little to no system testing has been performed, however, and performance/scale data is not yet available. Preliminary documentation is available.

In GigaVUE-FM you can control the level of logs created at each individual GCB for troubleshooting.

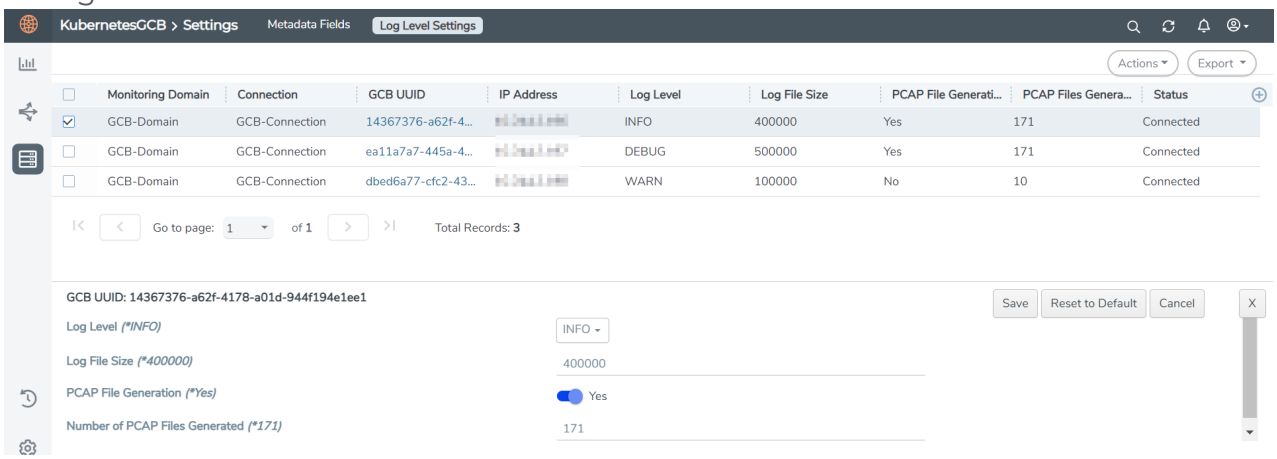
To view or edit the GCB log level settings:

1. In GigaVUE-FM, navigate to **Inventory > VIRTUAL > KubernetesGCB > Settings**, the **Settings** page appears.
2. From the **Settings** page, select **Log Level Settings** tab to view the list of GCB settings.

3. Select a GCB and select **Actions > View Configuration** to view the GCB log configuration.



4. Select a GCB and select **Actions > Edit Configuration** to edit the selected GCB log configuration.



NOTE: You can select multiple GCBs to modify the configuration with the same value.

Field	Description
Log Levels	Select one of the following: • DEBUG—fine-grained log information for application debugging • INFO—coarse-grained log information for highlighting application progress • WARN—log information of potentially harmful situations • ERROR—log information of the error events that allows the application to run continuously • FATAL—log information of very severe error events that presumably lead the application to abort.
Log File Size	Enter a value for the number of lines in the GCB log file.
PCAP File Generation	Select **Yes** to generate the PCAP file and select **No** to continue without the PCAP file.
Number of PCAP Files Generated	Enter a value for the number of PCAP files to be generated and stored on the GCB.

GCB for Cloud Object Storage

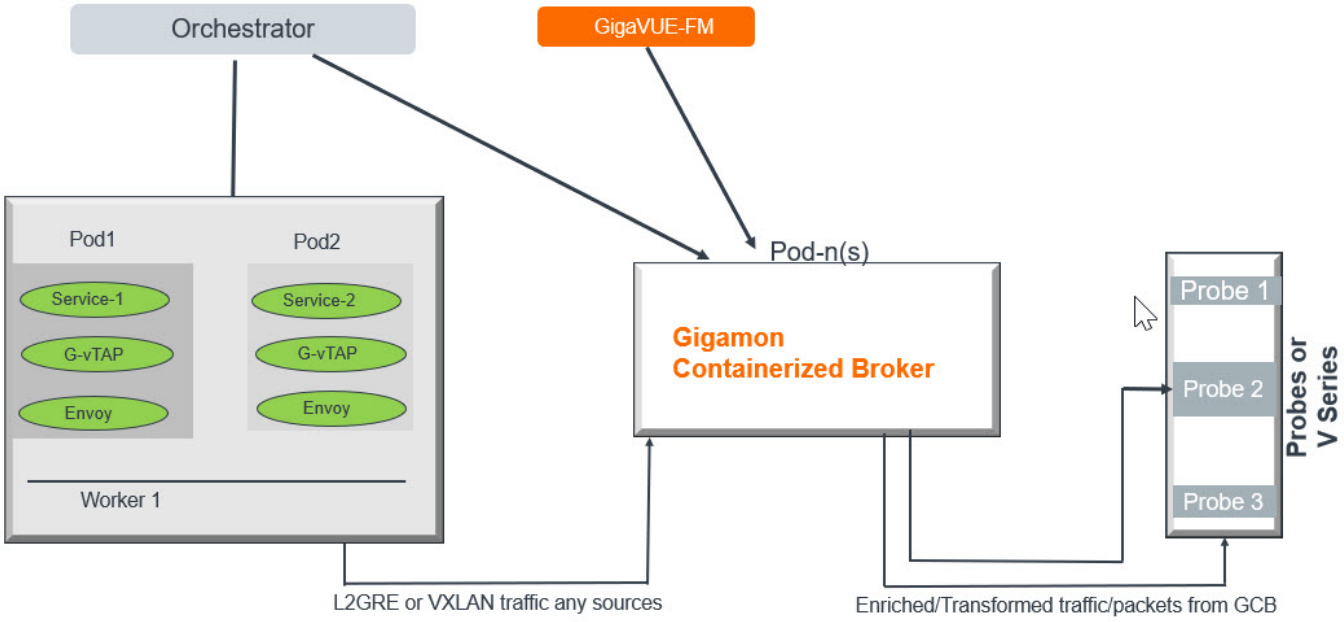
This chapter provides an overview of Gigamon Containerized Broker for cloud object storage and describes how to install and deploy G-vTAP Containers in your PODs.

Topics:

Architecture of GCB for Cloud Object Storage

GCB with GigaVUE-FM deployment

With GCB in its own POD, you can choose an orchestrator (other than GigaVUE-FM) like K8S to spin up/down the GCB pods.



During GCB initialization, the GCB Controller tries to connect with the GigaVUE-FM IP that you provided in the YAML file. GigaVUE-FM has a server certificate and GCB has a client certificate, so that GigaVUE-FM and GCB can identify the connection and traffic flow. GigaVUE-FM does not control the GCB spin up/down. The GCB parameter definition and deployment is performed through Kubernetes orchestrator and not by GigaVUE-FM.

Get Started with GCB for Cloud Object Storage

This section describes how to initiate GCB deployment with the required licenses and network requisites.

Refer to the following sections for details:

- Components of GCB for Cloud Object Storage
- License Information
- Network Requirements

Components of GCB for Cloud Object Storage

The Gigamon Containerized Broker for cloud object storage works with the following components:

- **GigaVUE® Fabric Manager (GigaVUE-FM)** is a web-based fabric management and orchestration interface that provides a single pane of glass visibility, management, and orchestration of both the physical and virtual traffic that form the GCB.
- **G-vTAP Container** is the Traffic Acquisition Component of Gigamon's Network Visibility Offering. It receives mirrored traffic from various Networking Infrastructures and overlays (VXLAN) them to Gigamon Containerized Broker.
- **GCB Controller** is the management component of GCB that controls the registration and deregistration with GigaVUE-FM. GCB Controller also sends the collected statistics of GCB and G-vTAP Containers to GigaVUE-FM.
- **GCB S3** is the storage service component of GCB that collects the mirrored packets from GCB Controller, converts to PCAP file and uploads it into Amazon S3.

License Information

All the G-vTAP instances connected to GCB periodically report the statistics to GCB. Then the GCB periodically reports the collective statistics of G-vTAPs and its own statistics to GigaVUE-FM for Volume-Based Licensing. GigaVUE-FM adds the required licensing tags into the Elasticsearch.

In the Volume-Based Licensing scheme, a license entitles specific applications on your devices to use a specified amount of total data volume over the term of the license. The distribution of the license to individual nodes or devices becomes irrelevant for Gigamon’s accounting purpose. GigaVUE-FM tracks the total amount of data processed by the various licensed applications and provides visibility into the actual amount of data, each licensed application is using on each node, and tracks the overuse if any.

Network Requirements

A security group defines the virtual firewall rules for your instance to control inbound and outbound traffic. When you launch GigaVUE-FM, Gigamon Containerized Broker, and G-vTAP Containers in your project, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic.

It is recommended to create a separate security group for each component using the rules and port numbers.

Direction	Type	Protocol	Port	CIDR	Purpose
Gigamon Containerized Broker deployed inside EKS worker node					
Inbound	HTTPS	TCP	443	Any IP address	Allows GCB Controller to communicate with GigaVUE-FM

Configure GCB for Cloud Object Storage

This section describes how to configure GCB in your environment. Refer to the following section for details.

- Deploy G-vTAP Containers
- Launch GigaVUE-FM
- Launch Gigamon Containerized Broker
- Store Traffic Data in S3 Bucket

Deploy G-vTAP Containers

Follow the instructions below to deploy G-vTAP Containers in your node:

1. In your Kubernetes orchestrator, enter the G-vTAP Container image name, commands and the required information in a YAML file. Following is the example data to be entered into your YAML file:

```
image: gigamon/gvtap-container: :<version>
#imagePullPolicy: Never
#imagePullPolicy: Always
#imagePullPolicy: IfNotPresent
command: ["/gvtap", "1", "eth0", "eth0", "10.9.0.216", "4789", "45"]
```

2. Using the YAML file, Kubernetes automatically downloads the defined G-vTAP Container and deploys in the selected PODs.

Launch GigaVUE-FM

The recent GigaVUE-FM image files can be downloaded from Gigamon Customer Portal. After fetching the image, upload and launch GigaVUE-FM inside or outside your VPC. For assistance, Contact Technical Support of Gigamon.

Launch Gigamon Containerized Broker

Follow the instructions below to deploy GCB in your node:

1. In your Kubernetes orchestrator, enter the GCB Controller and GCB S3 image name, commands and the required information in a YAML file. Following is the example data to be entered into your YAML file:

```
image: gigamon/gcb-s3:<version>
- command:
- gcb-s3
- <pkt_filter_type(ip|tcp|udp)>
- <i_iface: eth0, eth1>
- <s3_bucket_name>(Ex: gcb_s3_bucket)
- <s3_region>(Ex: us-east-2>)
- <AWS Account-ID>
- <max_pkt_per_pcap>
- <idle_timeout (in sec)>
- <stats_active (0/1)>
- <gcm port>
- <stats_interval(in sec)>
- <filtering rule>
- <gcb vxlan port>

image: gigamon/gcb-cntlr:<version>
- command:
- /gcb-cntlr
- <GigaVUE-FM IP>
- <PORT ID for GCB controller to communicate with GigaVUE-FM>
```

2. Using the YAML file, Kubernetes automatically downloads the defined GCB Controller and GCB S3. Then both are deployed in a new POD.
3. Connect the deployed G-vTAP Containers to the GCB installed in the same node.
4. Register GCB with the GigaVUE-FM launched inside or outside your VPC.

Once the GCB is registered with GigaVUE-FM, the GCB starts to collect the traffic from the G-vTAP Containers and periodically sends the heartbeats and statistics to GigaVUE-FM. For more information on GCB and GigaVUE-FM interaction, refer to GCB and GigaVUE-FM Interaction

Store Traffic Data in S3 Bucket

By default, the traffic information from GCB is saved into Amazon S3 bucket. All the parameters of the S3 bucket are defined in the yaml files.

The following are the S3 bucket parameters defined in yaml file:

Parameter	Description
s3_bucket_name	Name of the Amazon S3 bucket
s3_region	AWS region (Example: us-east-2>)
AWS Account-ID	ID of AWS user account
max_pkt_per_pcap	Maximum packets required to create a PCAP file
idle_timeout (in sec)	Idle time limit to create PCAP file without waiting to collect the maximum packets defined.

Follow the instructions below to store the traffic data from GCB to your Amazon S3 bucket.

1. Save the traffic data from the GCB as a PCAP file with the Server-Side Encryption technology.
2. Transfer and save the encrypted PCAP files to your Amazon S3 bucket.

NOTE: Naming convention of the PCAP file and the folder in S3 bucket are as follows:

- PCAP file name: **<AWS Accout ID>_pod_<POD IP>_YYYY_MM_DD_HH_mm_ss_<milliseconds>.pcap**
- S3 folder name: **[S3 bucket name]/account_id/MM-DD-YYYY/[file-name]/**

View GCB statistics in GigaVUE-FM

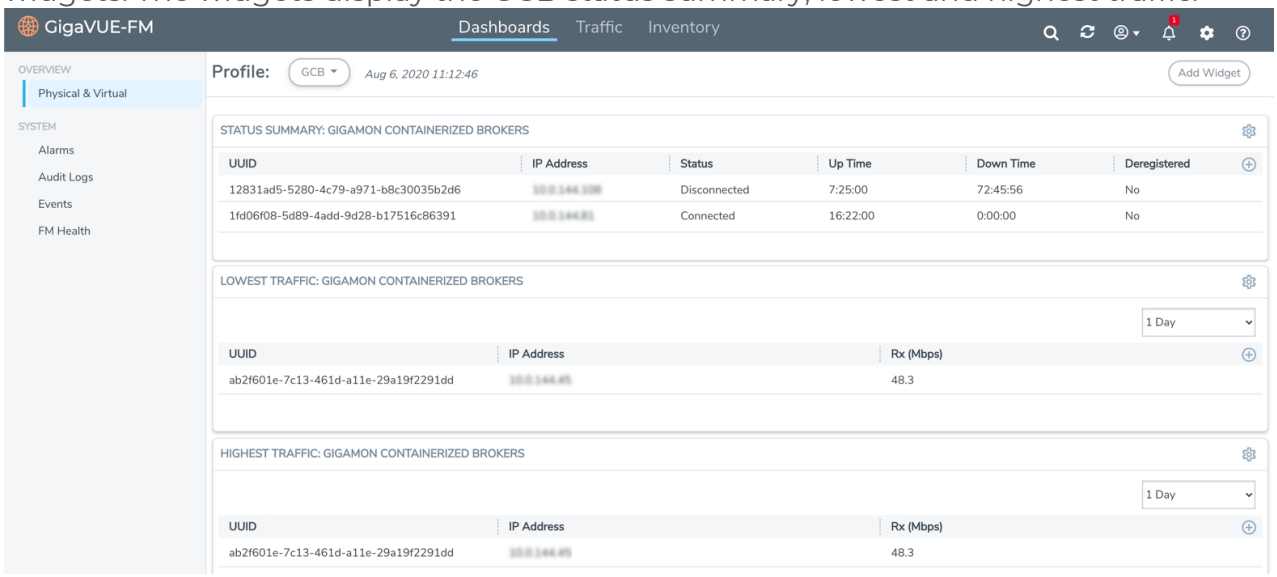
You can view the traffic information of GCB in GigaVUE-FM as the collective traffic from G-vTAPs and GCB are periodically transferred to GigaVUE-FM.

GigaVUE-FM dashboard displays the GCB statistics in the following widgets:

- Status Summary
- Lowest Traffic
- Highest Traffic

To view the GCB statistics in GigaVUE-FM:

1. On the top navigation bar, click **Dashboard**.
2. In the left navigation pane of the Dashboard page, click **Physical & Virtual**.
3. Click **Add Widget** and select Status Summary, Lowest Traffic, and Highest Traffic widgets. The widgets display the GCB status summary, lowest and highest traffic.



GCB Reference

This section provides additional references useful for GCB.

Configure mTLS Authentication

Mutual TLS (mTLS) authentication or two-way authentication refers to the two parties (GigaVUE-FM & GCB, and Pcapper & GCB) authenticating each other at the same time in an authentication protocol. mTLS can protect against adversarial attacks and ensure information integrity.

GigaVUE-FM supports mTLS (basic authentication) using the username and password. Proper certificates need to be installed on both GigaVUE-FM and your environment, as default generated certificates will not work with mTLS.

NOTE: During GigaVUE-FM upgrade, the files only with the **.crt** or **.key** under **/etc/pki/tls** extensions will be retained.

Configure mTLS Authentication in GigaVUE-FM

Follow the below steps to configure mTLS authentication in GigaVUE-FM:

1. Log in to the GigaVUE-FM CLI.
2. Ensure that you have the following certificates and keys in the **/home/User/certsAndKeys** directory:

   > NOTE: The names of the certificates and keys are configurable and can be changed. You must make sure that you use the same names in the configurations that follow.

   - **fmServerCertificate.pem**: public certificate file in PEM format to be used by GigaVUE-FM when acting as a server.
   - **fmServerCertificateKey.pem**: private key file in PEM format to be used by GigaVUE-FM when acting as a server.
   - **fmServerCACertificate.pem**: public certificate file in PEM format of the CA which issued the fmServerCertificate.pem to be used by GigaVUE-FM when acting as a server.

     > NOTE: fmServerCACertificate.pem certificate needs to be imported into client's TrustStore, including browser if it is not issued by one of the trusted CAs.

   - **fmClientCertificate.pem**: public certificate file in PEM format to be used by GigaVUE-FM when acting as a client.
   - **fmClientCertificateKey.pem**: private key file in PEM format to be used by GigaVUE-FM when acting as a client.
   - **fmClientCACertificate.pem**: public certificate file in PEM format of the CA which issued the fmClientCertificate.pem to be used by GigaVUE-FM when acting as a client.

   > NOTE: If the same certificate is used when GigaVUE-FM is a client and as a server, the three **fmServer*.pem** files will be the same as the three **fmClient*.pem** files.

3. Change to the directory where the above files are stored.

   ```
   cd /home/User/certsAndKeys
   ```

4. Add **fmClientCACertificate.pem** to the GigaVUE-FM trust store:

   ```
   sudo cp fmClientCACertificate.pem   /etc/pki/ca-trust/source/anchors/

   sudo update-ca-trust extract
   ```

5. Install the certificates and private key to make GigaVUE-FM act as a server.
   a. Backup the existing certificate and copy new FM certificate **fmServerCertificate.pem**.

      ```
      sudo cp /etc/pki/tls/certs/localhost.crt /etc/pki/tls/certs/BACKUP_
      localhost.crt
      ```

      ```
      sudo cp fmServerCertificate.pem /etc/pki/tls/certs/localhost.crt
      ```

   b. Backup the existing private key and copy new GigaVUE-FM key **fmServerCertificateKey.pem**

      ```
      sudo cp /etc/pki/tls/private/localhost.key
      /etc/pki/tls/private/BACKUP_localhost.key
      ```

      ```
      sudo cp fmServerCertificateKey.pem /etc/pki/tls/private/localhost.key
      .
      ```

   c. GigaVUE-FM uses a public key (cms.p12 file) to encrypt the Security Assertion Markup Language (SAML) messages. Since for mTLS to work, we need valid certificates installed in FM, generating a new public key using the following command:

      ```
      sudo openssl pkcs12 -export -name CMS -out /etc/gigamon/cms.p12 \
      ```

      ```
      -inkey /etc/pki/tls/private/localhost.key -in
      /etc/pki/tls/certs/localhost.crt -passout pass:cms123
      ```

6. Install the certificates and private key to make GigaVUE-FM act as a client.
   a. Copy new client certificate **fmClientCertificate.pem**.

      ```
      sudo cp fmClientCertificate.pem
      /etc/pki/tls/certs/fmClientCertificate.crt
      ```

   b. Copy new client key **fmClientCertificateKey.pem**.

      ```
      sudo cp fmClientCertificateKey.pem
      /etc/pki/tls/private/fmClientCertificateKey.key
      ```

   c. Copy new client CA public certificate **fmClientCACertificate.pem**.

      NOTE: This certificate needs to be imported into GigaVUE-FM Trust Store.

      ```
      sudo cp fmClientCACertificate.pem
      /etc/pki/tls/certs/fmClientCACertificate.crt
      ```

7. Generate KeyStore for GigaVUE-FM to act as a client
   a. Create a client certificate chain file.

      ```
      sudo cat /etc/pki/tls/certs/fmClientCACertificate.crt \

              /etc/pki/tls/certs/fmClientCertificate.crt \

              /etc/pki/tls/private/fmClientCertificateKey.key | sudo tee
      /etc/pki/tls/certs/fmClient.chain.crt > /dev/null
      ```

   b. Create a client certificate chain file in PKCS12 format.

      ```
      sudo openssl pkcs12 -export -in /etc/pki/tls/certs/fmClient.chain.crt
      \

      -out /etc/pki/tls/certs/fmClient.chain.p12 \

      -passout pass:changeit
      ```

   c. Create Java keystore

      ```
      sudo keytool -importkeystore -srckeystore
      /etc/pki/tls/certs/fmClient.chain.p12 \

      -srcstoretype pkcs12 \

      -destkeystore /etc/pki/tls/certs/fmClientJKS.crt \

      -storepass changeit
      ```

   d. Make the keystore readable.

      ```
      sudo chmod 644/etc/pki/tls/certs/fmClientJKS.crt
      ```

   e. Configure GigaVUE-FM load balancer functionality.

      ```
      cat /etc/pki/tls/certs/localhost.crt
      /etc/pki/tls/private/localhost.key > /etc/pki/tls/certs/localhost.pem

      curl -XPOST "localhost:4466/fmcs/configureLoadBalancer?pretty" -H
      "Content-Type: application/json" -d '{"custom_certificate" : true}'
      ```

   f. Restart Apache Web Server.

      ```
      sudo systemctl restart httpd
      ```

   g. Restart the GigaVUE-FM.

      ```
      sudo systemctl restart tomcat@cms.service
      ```

GigaVUE-FM is not responsible for any PKI or certificate management activities.

Configure mTLS Authentication in GCB

Follow the below steps to configure mTLS authentication in GCB:

> **NOTE:** Before you begin, you must generate the **ca_cert.pem**, **gcb_cert.pem** and **gcb_key.pem**certificates for FM-GCB mTLS configuration.

1. Copy the generated **ca_cert.pem**, **gcb_cert.pem** and **gcb_key.pem** certificates that you generated, to a folder.
2. Create a secret using mTLS for GCB in Kubernetes by using the below command and giving respective path to each file:

   ```
   kubectl create secret generic <secret-name> --from-file=gcb-ca-root-cert=<path to file> --from-file=gcb_cert=<path to file> --from-file=gcb-pvt-key=<path to file>
   ```

3. Use the above created secret in the following snippet from gcb-cntlr YAML file.

   ```
   - mountPath: /etc/gcbcerts
   mountPropagation: None
   name: gcb-tls
   volumes:
   - name: gcb-tls
   secret:
   secretName: gcb-tls
   ```

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 Gigamon Community

Documentation

This table lists all the guides provided for GigaVUE Cloud Suite 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 Cloud Suite 5.15 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, rack-mount, connect, and initially configure ports the respective GigaVUE Cloud Suite devices; reference information and specifications for the respective GigaVUE Cloud Suite devices
G-TAP A Series 2 Installation Guide
GigaVUE-HC1 Hardware Installation Guide
GigaVUE-HC2 Hardware Installation Guide
GigaVUE-HC3 Hardware Installation Guide
GigaVUE M Series Hardware Installation Guide
GigaVUE TA Series Hardware Installation Guide
GigaVUE-OS Installation Guide for DELL S4112F-ON

GigaVUE Cloud Suite 5.15 Hardware and Software Guides
Software Installation and Upgrade Guides
GigaVUE-FM Installation, Migration, and Upgrade Guide
GigaVUE-OS Upgrade Guide
Administration
GigaVUE Administration Guide covers both GigaVUE-OS and GigaVUE-FM
Fabric Management
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
Cloud Configuration and Monitoring how to configure the GigaVUE Cloud Suite components and set up traffic monitoring sessions for the cloud platforms
GigaVUE V Series Quick Start Guide
GigaVUE Cloud Suite for AWS–GigaVUE V Series 2 Guide
GigaVUE Cloud Suite for AWS–GigaVUE V Series 1 Guide
GigaVUE Cloud Suite for Azure Guide
GigaVUE Cloud Suite for OpenStack Guide
Gigamon Containerized Broker Guide
GigaVUE Cloud Suite for VMware—GigaVUE V Series Guide
GigaVUE Cloud Suite for AnyCloud Guide
GigaVUE Cloud Suite for Kubernetes Guide
GigaVUE Cloud Suite for Nutanix Guide
GigaVUE Cloud Suite for VMware—GigaVUE-VM Guide
GigaVUE Cloud Suite for AWS Secret Regions Guide
Reference
GigaVUE-OS CLI Reference Guide library of GigaVUE-OS CLI (Command Line Interface) commands used to configure and operate GigaVUE H Series and TA Series devices
GigaVUE-OS Cabling Quick Reference Guide guidelines for the different types of cables used to connect Gigamon devices

GigaVUE Cloud Suite 5.15 Hardware and Software Guides
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)
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 & 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.
GigaVUE-OS H-VUE Online Help provides links the online documentation.

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 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 Gigamon Community

The Gigamon 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 Gigamon 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.
- Submit and vote on feature enhancements and share product feedback. (Customers only)
- Open support tickets (Customers only)
- Download the latest product updates and documentation (Customers only)

The Gigamon 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)