Gigamon®

Universal Cloud Tap-Container Deployment Guide

Product Version: 6.14

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
6.14	1.0	07/10/2026	The original release of this document with 6.14.00 GA.

Contents

Overview of Universal Cloud Tap - Container

Universal Cloud Tap - Container (UCT-C), earlier known as Universal Container Tap (UCT), is a containerized component deployed as a DaemonSet. This ensures that it runs on every worker node within a cluster. UCT-C provides network broker features in a containerized form and can perform traffic acquisition, basic filtering, and tunneling support. It is deployed as a Pod in the given worker node where the workloads run.

UCT-C is deployed by the Kubernetes orchestrator and not by GigaVUE-FM. The traffic acquisition process is initiated by UCT-C.

Following are the modules implemented in UCT-C:

- **Traffic Acquisition** - UCT-C supports traffic acquisition by replicating the traffic from the worker pods.
- **Filtering Module** - UCT-C provides basic filtering based on 5-Tuple. The filtering configuration is pushed by GigaVUE-FM.
- **Tunneling Modules** - UCT-C supports L2GRE, VXLAN, and TLS-PCAPng tunneling to send the tapped traffic to the GigaVUE V Series Nodes or tools.

Architecture of Universal Cloud Tap - Container

UCT-C enables the capture of network traffic directly from Kubernetes workload pods and facilitates its redirection to designated tunnel destinations, such as V Series appliances or monitoring and security tools.

Kubernetes deploys the UCT-C Controller as a single-instance deployment (replica count: 1). It also creates a Controller Service to enable communication with the controller.

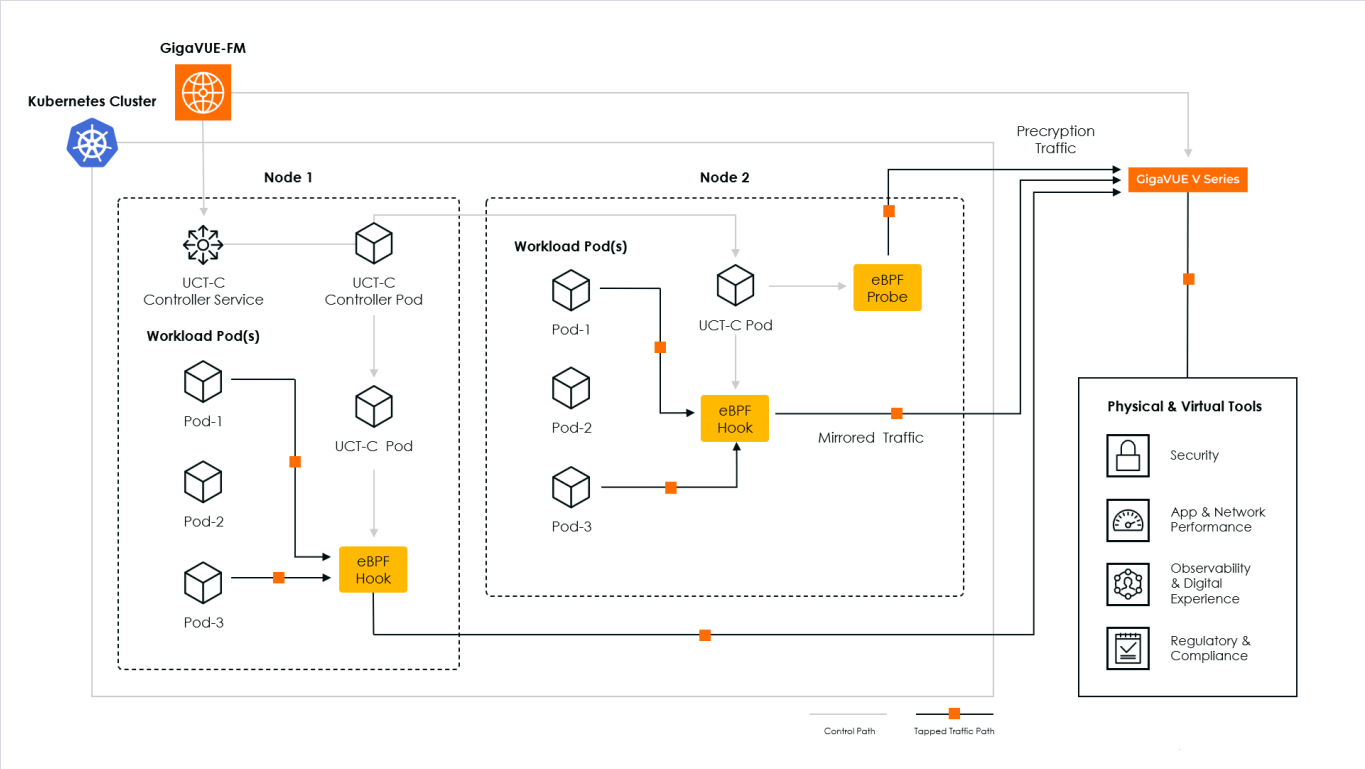
The Universal Cloud Tap - Container works with the following components:

- **GigaVUE-FM Fabric Manager** is a browser-based fabric management and orchestration interface that provides a centralized orchestration and visibility across physical and virtual fabrics.
- **UCT-C Tap** is the primary UCT-C module that collects the workload traffic, filters it, and tunnels the filtered traffic directly to the tools or through the GigaVUE V Series Nodes. It also sends the traffic policy statistics and heartbeats to the UCT-C Controller. UCT-C Tap must run as a **privileged pod**.

  NOTE: UCT-C uses eBPF (extended Berkeley Packet Filter) to tap traffic from user pods. eBPF runs on the Linux kernel and requires privileged pod permission in Kubernetes. UCT-C Tap pods require SYS_ADMIN and NET_ADMIN privileges to attach eBPF Hooks, run commands in other namespaces, and run low-level networking commands.

- **UCT-C Controller** is the management component of UCT-C that controls and communicates with UCT-C Tap. UCT-C Controller collects the data from UCT-C Taps and sends the collected statistics and heartbeats to GigaVUE-FM.

The following diagram illustrates the various components involved in the UCT-C solution.



The UCT-C Controller acts as the central communication link between GigaVUE-FM and UCT-C Taps, managing registration, policy deployment, data collection, and statistics reporting. The process begins with the UCT-C Controller registering with GigaVUE-FM, after which the UCT-C Taps are also registered through the Controller. Once registered, GigaVUE-FM communicates with the UCT-C Taps exclusively through the UCT-C Controller, ensuring a structured flow of configuration and data.

When GigaVUE-FM deploys traffic policies, the controller receives and distributes them to the connected UCT-C Taps, instructing them on how to filter and process network traffic. The filtered network packets are then either tunneled directly to connected monitoring and security tools or sent through GigaVUE V Series Nodes deployed within a supported GigaVUE Cloud Suite environment.

The UCT-C Controller also manages ongoing communication and monitoring by collecting data from the UCT-C Taps, including traffic statistics and system health information. It consolidates these details and transmits them back to GigaVUE-FM, ensuring continuous

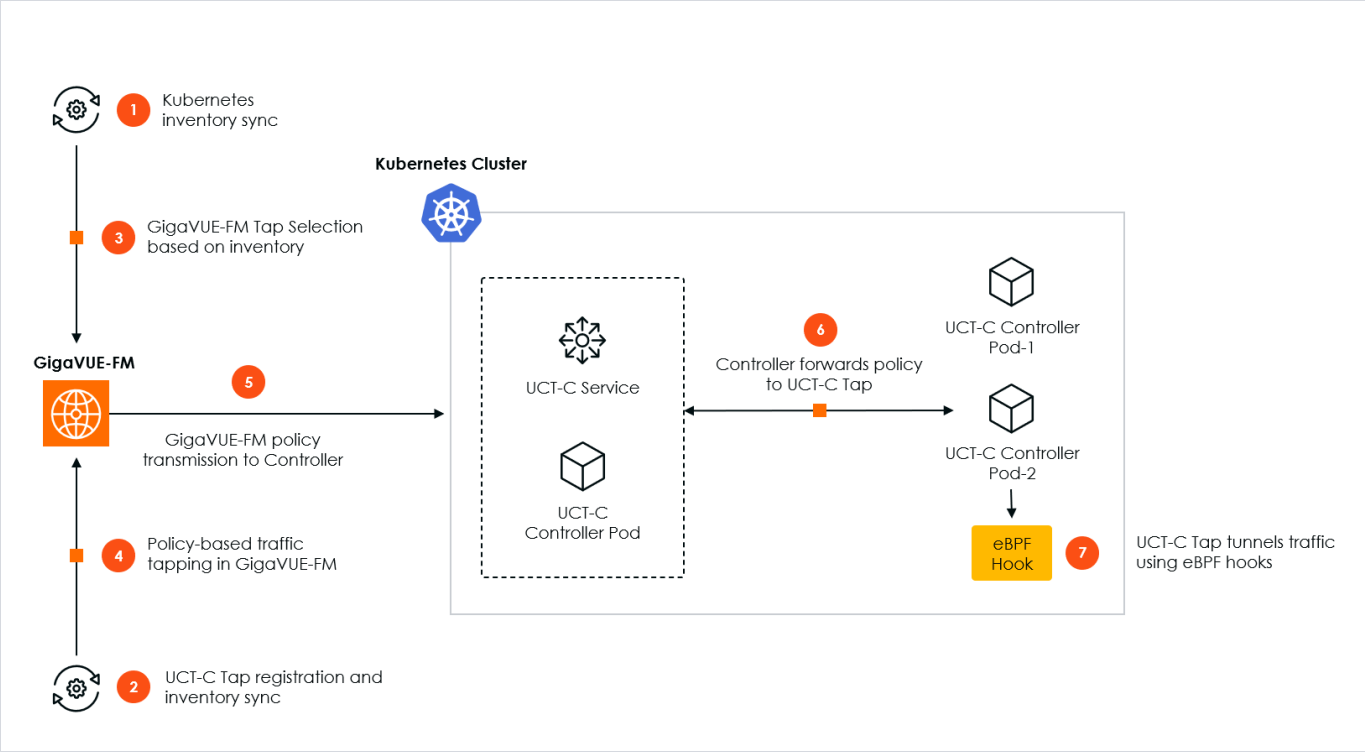
visibility into network activity. Additionally, the controller sends periodic heartbeats to confirm the status and availability of UCT-C Taps, enabling GigaVUE-FM to monitor the system's overall health and performance.

The UCT-C solution can tap mirrored and Precryption traffic (refer to the architecture diagram above). UCT-C provides multiple tunneling options to send the acquired traffic to external systems for further processing or analysis. The supported tunneling types are:

- **Layer 2 Generic Routing Encapsulation (L2GRE)**: A tunneling protocol that encapsulates Layer 2 traffic to transport it over an IP network.
- **Virtual Extensible LAN (VXLAN)**: A network virtualization technology that encapsulates Layer 2 Ethernet frames inside Layer 3 UDP packets. It is typically used to create a virtualized Layer 2 network that runs over a Layer 3 infrastructure, allowing for better scalability and flexibility in cloud environments.
- **Transport Layer Security - PCAPng (TLS-PCAPng)**: A secure tunneling method that captures traffic in a standardized PCAPng format, encrypted using TLS, ensuring the confidentiality and integrity of the data during transmission.

These tunneling protocols allow the UCT-C to send the captured traffic to the GigaVUE V Series Nodes or external network monitoring tools, ensuring that all network traffic data is securely transmitted and available for analysis.

Traffic Tapping in Kubernetes Using GigaVUE-FM



1. **Kubernetes inventory sync:**

When the Controller starts, it gathers inventory information about the Kubernetes cluster in which it runs, including data on worker nodes, Kubernetes services, and Pods. The controller then sends this information to GigaVUE-FM. It also registers with the Kubernetes cluster to receive real-time notifications of any changes in the cluster's inventory. As changes occur, the controller pushes these updates to GigaVUE-FM.

2. **UCT-C Tap Registration and Inventory Sync:**

   When a UCT-C Tap starts on a node, it sends its registration message to the controller, which then forwards it to GigaVUE-FM. This way, GigaVUE-FM knows which UCT-C Tap runs on which worker node.

3. **GigaVUE-FM Tap Selection Based on Inventory:**

   Access to inventory information and knowing which UCT-C Taps are running on which nodes allow GigaVUE-FM to apply configuration policies and rules. You can choose the pods to tap for traffic based on Kubernetes resources such as nodes, namespaces, and pods.

4. **Policy-Based Traffic Tapping in GigaVUE-FM:**

   When a user configures a policy for tapping, GigaVUE-FM determines the list of worker pods that need to be tapped based on the Source Selection criteria. It also identifies the nodes on which these pods are running. Subsequently, GigaVUE-FM requests the controller to configure the UCT-C Tap on each node to tap traffic from the specified worker pods.

5. **GigaVUE-FM Policy Transmission to Controller:**

   When GigaVUE-FM sends this policy to the controller.

6. **Controller Forwards Policy to UCT-C Tap:**

   The controller forwards the policy configuration to the UCT-C Tap, which is responsible for implementing the policy. The UCT-C Tap then applies the policy to tap traffic from the specified worker pods.

7. **UCT-C Tap Tunnels Traffic using eBPF Hooks:**

   After this, the UCT-C Tap adds eBPF hooks at the TC layer to get the traffic and tunnel it to the designated endpoint specified in the policy. Once this process is complete, traffic to and from these pods is tunneled to the destination outlined in the policy.

Communication between UCT-C and GigaVUE-FM

This section describes the different communication channels between UCT-C and GigaVUE-FM.

Refer to the following section:

- Controller to GigaVUE-FM Communication
- GigaVUE-FM to Controller Communication
- Controller to Tap Communication
- Tap to Controller Communication

Controller to GigaVUE-FM Communication

The Controller and the GigaVUE-FM communicate using TLS or mTLS protocols. The process includes the following responsibilities:

- Identifies how to reach GigaVUE-FM.
- Connects to GigaVUE-FM to register itself.
- Reports the initial inventory to GigaVUE-FM.
- Reports any changes in the inventory to GigaVUE-FM.
- Forwards UCT-C Tap registration information to GigaVUE-FM.
- Maintains a record of each tap in its memory table.
- Reports the heartbeat of each UCT-C Tap instance to GigaVUE-FM.
- Reports Volume Based Licensing (VBL) statistics to GigaVUE-FM.
- Reports Pod/Policy statistics to GigaVUE-FM.

NOTE: Data packets from the selected source to the tunnels do not pass through the Controller

GigaVUE-FM to Controller Communication

GigaVUE-FM to Controller communication utilizes TLS or mTLS. The process includes the following responsibilities:

- Sends Policy configuration requests for UCT-C Pod(s).
- Registers UCT-C instances with the Controller for management
- Secures the entire communication from TLS and mTLS.

Controller to Tap Communication

The communication between the Controller and the Tap utilizes a secure channel, ensuring it is encrypted. The process includes the following responsibilities:

- Forwards policy configuration requests received from GigaVUE-FM to the Tap.
- Secure the communication channel.
- Pushes Traffic Forwarding and Routing configuration.
- Sends Traffic Filtering and Transformation updates.

Tap to Controller Communication

The communication between the Tap and the Controller utilizes a secure channel, ensuring it is encrypted. Each tap looks up the Controller Kubernetes Service. Taps do not communicate directly with GigaVUE-FM. All communications are routed through the

Controller.

The following actions occur during this process:

- Registers with GigaVUE-FM via the Controller.
- Sends heartbeat requests.
- Sends Volume Based Licensing (VBL) statistics.
- Sends pod or policy statistics.

Traffic Acquisition using UCT-C

UCT-C captures traffic from multiple container pod instances. It then forwards the traffic securely to a GigaVUE V Series Node. The V Series Node processes the traffic using advanced tools, including GigaSMART features. These tools improve network visibility, performance monitoring, and security analysis.

For details on Traffic Acquisition using UCT-C, refer to the following topics:

- Precryption
- Secure Tunnels

Precryption

License: Precryption requires a **SecureVUE Plus** license.

Gigamon Precryption™ technology[1] provides you clear-text visibility into encrypted network traffic without the need for traditional decryption. It works across virtual, cloud, and container environments, helping you get the full security stack without added complexity

In this section:

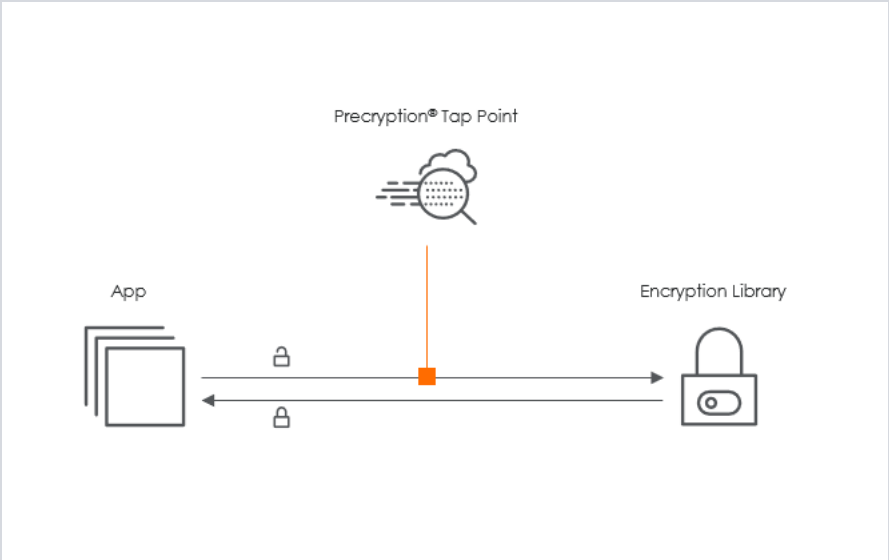
- How Gigamon Precryption Technology Works
- Why Gigamon Precryption
- Key Features
- Key Benefits
- Precryption Technology on Single Node
- Precryption Technology on Multi-Node
- Supported Platforms

[1]**Disclaimer**: The Precryption feature allows you to capture decrypted traffic from both virtual machine (VM) and container-based environments. After capturing the traffic using (via UCT-C or UCT-V), you can send to the V Series product for further processing. You can choose to secure this traffic using encrypted tunnels between the capture point and the V Series. This option helps protect sensitive data during transit. If you don't enable encrypted tunnels, the captured (decrypted) traffic remains in plain text while moving between the source and the V Series—introducing potential exposure risks. Please note that the feature behavior and security options may change over time. Stay informed about updates to ensure you use the latest protections. By using this feature, you acknowledge and accept the current limitations and potential risks associated with the transmission of decrypted traffic.

- Prerequisites

How Gigamon Precryption Technology Works

Precryption technology leverages built-in Linux functionality to copy communications between the application and the encryption library, such as OpenSSL.



Key Highlights

- Captures network traffic in plain text, either before the system encrypts it or after it decrypts it.
- Does not change how encryption or transmission works.
- Avoids proxies, retransmissions, and “break-and-inspect” steps. Instead, it sends the plaintext copy to the Gigamon Deep Observability Pipeline, where tools can optimize, transform, and forward the traffic as needed.
- Runs on GigaVUE® Universal Cloud Tap (UCT) and supports hybrid and multi-cloud environments, including on-prem and virtual platforms.
- Runs independently of your applications, so you don’t need to change your development lifecycle.

Why Gigamon Precryption

GigaVUE Universal Cloud Tap with Precryption technology is a lightweight, friction-free solution that eliminates blind spots present in modern hybrid cloud infrastructure.

Precryption helps you:

- Improve visibility for East-West traffic into virtual, cloud, and container platforms
- Delivers unobscured visibility into all encryption types, including TLS 1.3, without managing and maintaining decryption keys.
- Manages compliance with IT organizations, keeps communications private, architects a Zero Trust foundation, and boosts security-tool effectiveness by a factor of 5x or more.

Key Features

The following are the key features of this technology:

- Plain text visibility into communications with modern encryption (TLS 1.3, mTLS, and TLS 1.2 with Perfect Forward Secrecy).
- Plain text visibility into communications with legacy encryption (TLS 1.2 and earlier).
- Non-intrusive traffic access without agents running inside container workloads.
- Elimination of expensive resource consumption associated with traditional traffic decryption.
- Elimination of key management required by traditional traffic decryption.
- Zero performance impact based on cipher type, strength, or version.
- Support across hybrid and multi-cloud environments, including on-prem, virtual, and container platforms.
- Keep private communications private across the network with plaintext threat activity delivered to security tools.
- Integration with Gigamon Deep Observability Pipeline for the full suite of optimization, transformation, and brokering capabilities.

Key Benefits

The following are the key benefits of this technology:

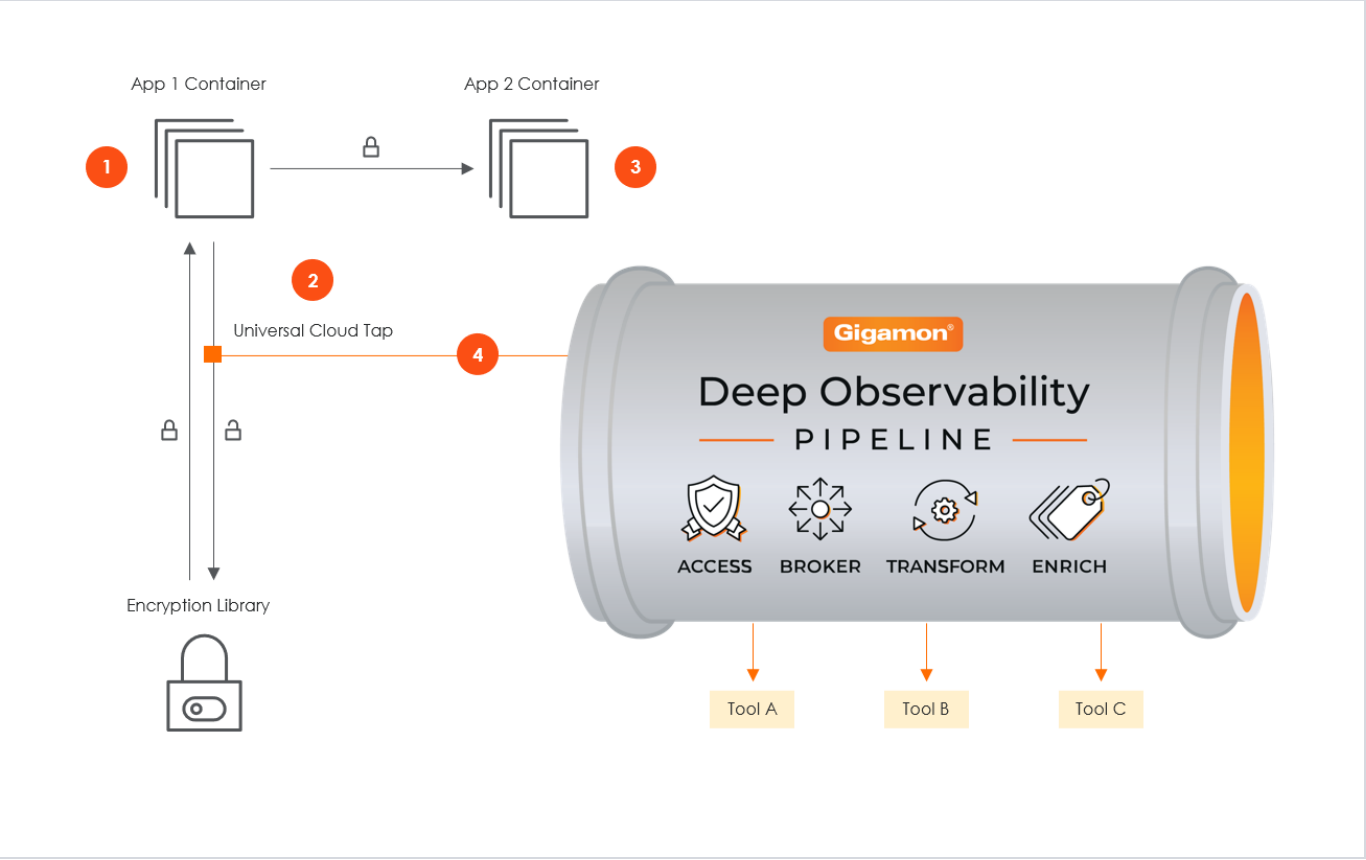
- Eliminates blind spots for encrypted East-West (lateral) and North-South communications, including traffic that may not cross firewalls.
- Monitors application communications with an independent approach that enhances development team velocity.
- Extends security tools' visibility to all communications, regardless of encryption type.
- Achieves maximum traffic tapping efficiency across virtual environments.
- Leverages a 5–7x performance boost for security tools by consuming unencrypted data.
- Supports a Zero Trust architecture founded on deep observability.
- Maintains privacy and compliance adherence associated with decrypted traffic management.

How Gigamon Precryption Technology Works

This section explains how Precryption technology works on single nodes and multiple nodes in the following sections:

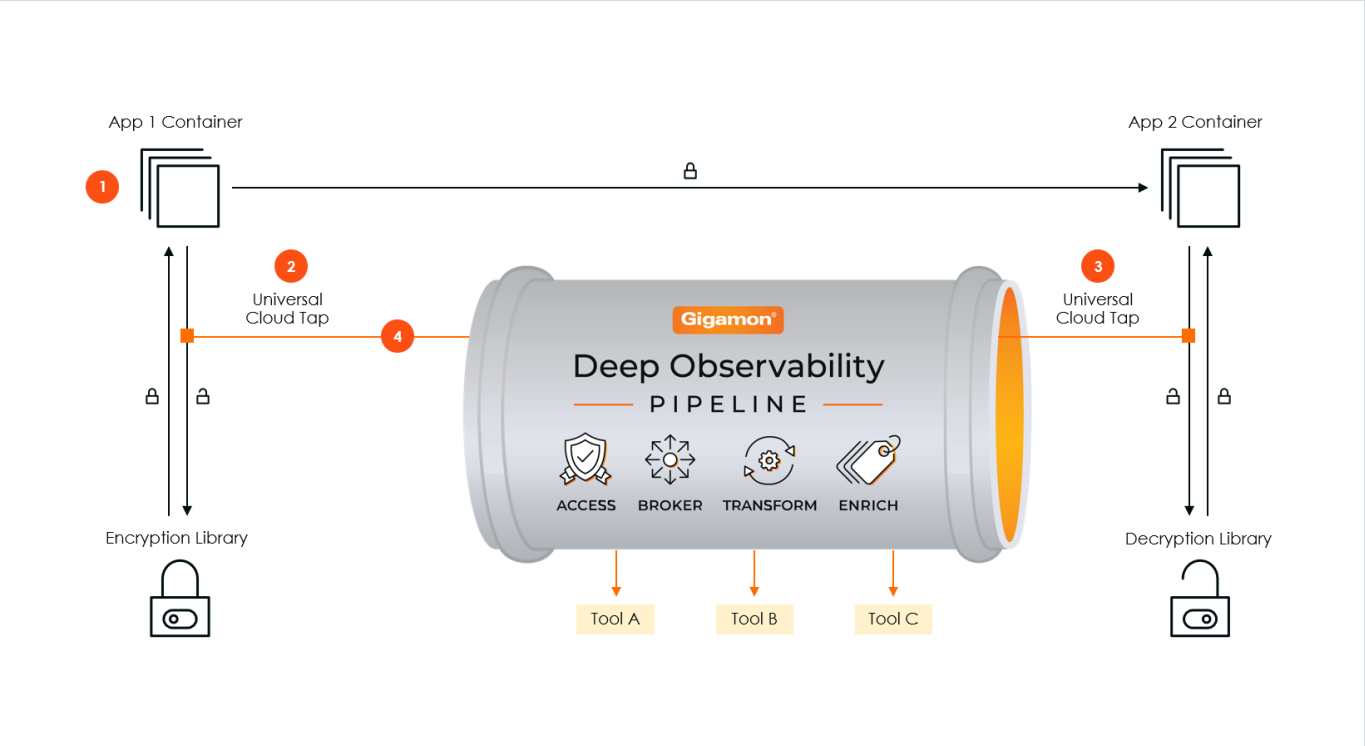
- Precryption Technology on Single Node
- Precryption Technology on Multi-Node

Precryption Technology on Single Node



1. An application uses an encryption library, such as OpenSSL, to encrypt a message.
2. GigaVUE Universal Cloud Tap (UCT), enabled with Precryption technology, gets a copy of this message before it is encrypted on the network.
3. The encrypted message is sent to the receiving application with unmodified encryption—no proxy, no re-encryption, no retransmissions.
4. GigaVUE UCT creates packet headers as needed, encapsulates them in a tunnel, and forwards them to GigaVUE V Series in the deep observability pipeline.
5. Gigamon optimizes, transforms, and delivers data to tools without further decryption.

Precryption Technology on Multi-Node



1. An application uses an encryption library, such as OpenSSL, to encrypt a message.
2. GigaVUE Universal Cloud Tap (UCT), enabled with Precryption technology, gets a copy of this message before it is encrypted on the network
3. Optionally, GigaVUE UCT enabled with Precryption can also acquire a copy of the message from the server end after the decryption.
4. GigaVUE UCT creates packet headers as needed, encapsulates them in a tunnel, and forwards them to GigaVUE V Series in the deep observability pipeline.
5. Gigamon optimizes, transforms, and delivers data to tools without further decryption.

Supported Platforms

VM environments: Precryption™ is supported on the following VM platforms that support UCT-V:

Platform Type	Platform
Public Cloud	• AWS • Azure • GCP (via Third Party Orchestration) • OCI (via Third Party Orchestration)
Private Cloud	• OpenStack • VMware ESXi (via Third Party Orchestration only)

Platform Type	Platform
	• VMware NSX-T (via Third Party Orchestration only) • Nutanix (via Third Party Orchestration only)

Container environments: Precryption™ is supported on the following container platforms that support UCT-C:

Platform Type	Platform
Public Cloud	• EKS • AKS • GKE • OKE
Private Cloud	• OpenShift • Native Kubernetes (VMware)

Prerequisites

Points to Note

- OpenSSL version 1.0.2, version 1.1.0, version 1.1.1, and version 3.x.
- For UCT-C, worker pods should always have libssl installed to ensure that UCT-C Tap can tap the Precryption packets from the worker pods whenever libssl calls are made from the worker pods.
- For GigaVUE-FM, add port 5671 in the security group to capture the statistics.
- In security group settings on the UCT-V Controller, enable Port 9900 to receive the statistics information from UCT-V.
- For UCT-C, add port 42042 and port 5671 to the security group.
- Precryption works only on Linux systems running Kernel version 4.18 or later.

License Prerequisite

- Precryption™ requires a SecureVUE Plus license.

Supported Kernel Version

Precryption is supported on kernel versions 4.18 and above, including 5.4+ kernels, across all Linux and Ubuntu operating systems. For the Kernel versions below 5.4, refer to the following table:

Kernel-Version	Operating System
4.18.0-193.el8.x86_64	RHEL release 8.2 (Ootpa)
4.18.0-240.el8.x86_64	RHEL release 8.3 (Ootpa)
4.18.0-305.76.1.el8_4.x86_64	RHEL release 8.4 (Ootpa)

Kernel-Version	Operating System
4.18.0-348.12.2.el8_5.x86_64	RHEL release 8.5 (Ootpa)
4.18.0-372.9.1.el8.x86_64	RHEL release 8.6 (Ootpa)
4.18.0-423.el8.x86_64	RHEL release 8.7 Beta (Ootpa)
4.18.0-477.15.1.el8_8.x86_64	RHEL release 8.8 (Ootpa)
5.3.0-1024-kvm	Ubuntu 19.10
4.18.0-305.3.1	Rocky Linux 8.4
4.18.0-348	Rocky Linux 8.5
4.18.0-372.9.1	Rocky Linux 8.6
4.18.0-425.10.1	Rocky Linux 8.7
4.18.0-477.10.1	Rocky Linux 8.8
4.18.0-80.el8.x86_64	CentOS 8.2
4.18.0-240.1.1.el8_3.x86_64	CentOS 8.3
4.18.0-305.3.1.el8_4.x86_64	CentOS 8.4
4.18.0-408.el8.x86_64	CentOS 8.5

For more details, refer to Gigamon TV.

Notes

- Refer to *Configure Precryption in UCT-V* topic in the respective GigaVUE Cloud Suite Guides for details on how to enable Precryption™ in VM environments.
- Refer to Configure Precryption in UCT-C for details on how to enable Precryption™ in container environments.

Secure Tunnels

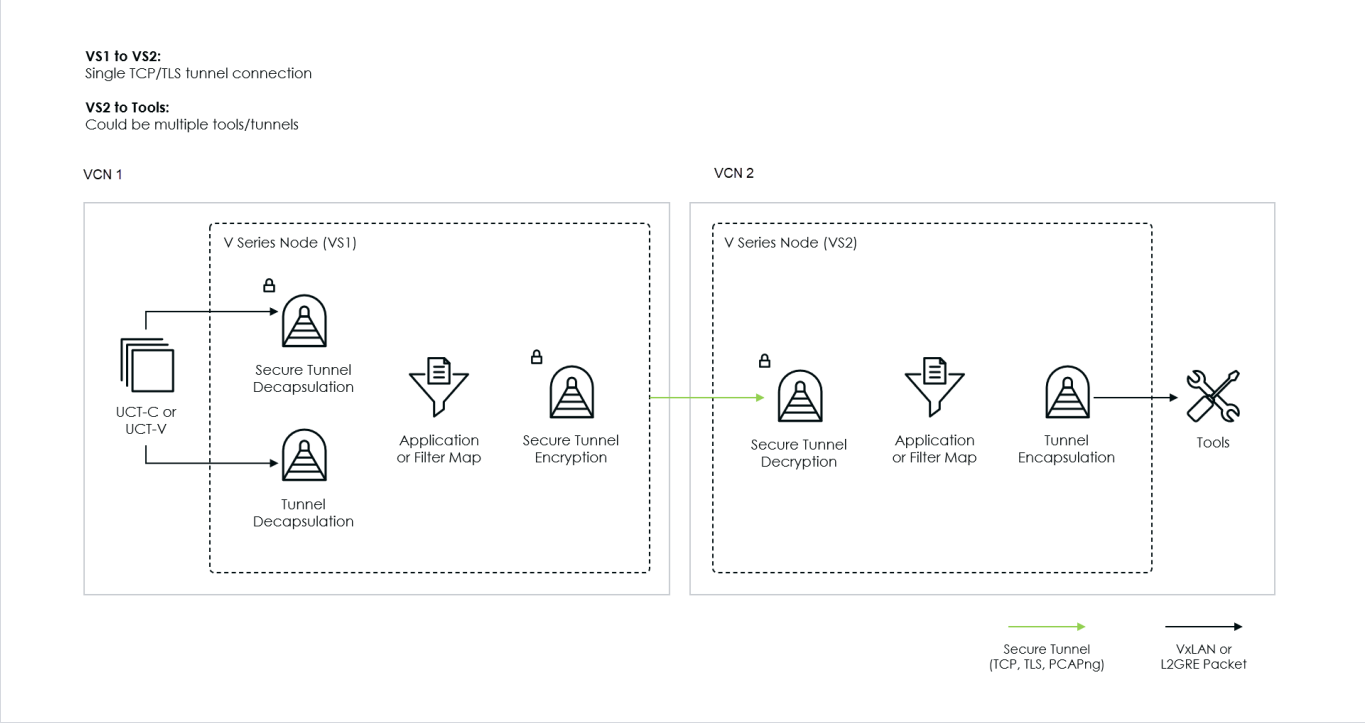
Secure Tunnels securely transfer the cloud-captured packets on UCT-V and UCT Container to a GigaVUE V Series Node.

- The data from UCT-C is encapsulated in PCAPng format.
- The encrypted data is sent over a TLS connection to a GigaVUE V Series Node.

Secure Tunnels can also transfer the captured packets from a GigaVUE V Series Node to another GigaVUE V Series Node or GigaVUE HC Series.

In the case of GigaVUE V Series Node to GigaVUE V Series Series node, the traffic from the GigaVUE V Series Node 1 is encapsulated using PCAPng format and transported to GigaVUE V Series Node 2, where the traffic is decapped. The secure tunnels between a V Series Node and a V Series Node have multiple use cases.

The GigaVUE V Series Node decapsulates and processes the packet as per the configuration. You can send the decapsulated packet to the application, such as De-duplication, Application Intelligence, Load balancer, and tool. The Load Balancer on this node can send the packets to multiple V Series Nodes. In this case, the packets can be encapsulated again and sent over a secure tunnel.



Supported Platforms

Secure Tunnels are supported on:

- OpenStack
- Azure
- AWS
- VMware NSX-T (only for Third Party Orchestration)
- VMware ESXi (only for Third Party Orchestration)
- Nutanix (only for Third Party Orchestration)
- Google Cloud Platform (only for Third Party Orchestration)

For information about how to configure secure tunnels, refer to the section Configure Secure Tunnels in UCT-C.

UCT-C Deployment Overview

Universal Cloud Tap - Container deployment follows a set of steps that include planning and preparation before the actual solution deployment. This section provides an overview of the activities associated with UCT-C deployment. The top-level deployment activities are:

Plan your Deployment	Address Prerequisites	Deploy UCT-C Solution
• Overview of Universal Cloud Tap - Container • Architecture of Universal Cloud Tap - Container • Traffic Acquisition using UCT-C • Security Requirements • Getting UCT-C Container Image • Uploading images in the local repository	• License Information • Supported Platforms for UCT-C • Network Ports Requirements • Compute Requirements • Kernel and CPU Requirements for Universal Cloud Tap - Container	• Launch GigaVUE-FM • Deploy UCT-C Solution in Kubernetes • Post Deployment • Configure UCT-C Solution using GigaVUE-FM • Configure UCT-C Features

UCT-C Deployment Planning

Before deploying UCT Container,check that you meet all requirements. Make sure your setup follows security rules and fits your infrastructure.

The sections below help you prepare. They show you how to get the UCT Container container image, set up security, and upload the image to a local repository. This makes deployment easier across your systems:

- Security Requirements
- Getting UCT-C Container Image
- Uploading images in the local repository

Security Requirements

Kubernetes Service Account

To create a Service Account with privileged access for tapping, run the following commands:

```
kubectl create ns uctc
kubectl create sa gigamon -n uctc
```

Use the Gigamon service account in uctc-tap.yaml that allows UCT-C to appear as a privileged pod.

If you are using the OpenShift Platform, refer to the following sections:

- **Using YAML**: In YAML deployment, for the Red Hat OpenShift Container Platform, use the following command:

```
oc adm policy add-scc-to-user -z gigamon privileged -n uctc
```

- **Using Helm**: In a Helm deployment for the Red Hat OpenShift Container Platform, set the create value to **True** under securityContextConstraints in values.yaml. This configuration generates a customized Security Context Constraint (SCC) with the necessary permissions required for deploying the UCT-C solution on OpenShift.

```
securityContextConstraints:
create: True
name: "gigamon"
```

NOTE: Security Context is not required in other platforms.

Access and Permissions Required for Deployment

To deploy the solution, you should have the below permissions:

- If you use standard ports like 443 for Controller to GigaVUE-FM communication, ensure to launch the Controller with privileged access.
- You should have Privileged user access since UCT-C Tap pods require privileged access for Mirroring or Precryption.

Getting UCT-C Container Image

You should create a Secret with registry credentials to pull the image from the Gigamon Software portal to your local server.

NOTE: This is optional if the images are in a local registry that is not password or token protected.

Run the following command to create the required secret:

```
oc create secret generic gigamon --from-file=.dockerconfigjson=<file_name> --
type=kubernetes.io/dockerconfigjson -n uctc
```

Uploading images in the local repository

Every release build in the Gigamon software portal includes UCT Container and Tap images.

To upload,

1. Download the UCT-C images from the Gigamon Software portal to your local server.
2. Untar the UCT-C image file and extract the images.
3. Upload the following files to your local repository:
   - **gigamon.gigavue-uctc-cntlr-image.tar**
   - **gigamon.gigavue-uctc-tap-image.tar**

You can use this file name in your deployment files.

NOTE: Contact Technical Support or Contact Sales for information on downloading the respective UCT-C build from the Gigamon software portal.

UCT-C Deployment Prerequisites

Before you deploy the UCT-C Solution with GigaVUE-FM, make sure you meet the required licensing, infrastructure, and technical knowledge requirements.

- **Kubernetes Knowledge:** You need the You need a basic understanding of Kubernetes to deploy and manage UCT-C. To learn more, refer to Kubernetes-basics.
- **Knowledge of Helm Charts:** Use Helm to install and manage UCT-C components. Helm charts bundle pre-configured Kubernetes resources and are included with the UCT-C image. Download and extract the UCT-C image to access the charts and README files. For detailed deployment steps, refer to Helm Charts.
- **Familiarity with Containerized workloads:** You need to understand how containerized workloads operate. UCT-C interacts with pods to acquire traffic and forward it for analysis. Make sure you know how to manage container networking, set resource limits, configure security, and handle dynamic scaling. These elements help UCT-C operate efficiently without disrupting workloads.

For detailed information on the prerequisites required for a successful UCT-C deployment, refer to the following sections:

- License Information
- Supported Platforms for UCT-C
- UCT-C Resource Requirements

License Information

All the UCT-C Taps deployed in your environment periodically report the statistics to UCT-C Controller. Then, the UCT-C Controller periodically reports the collective statistics of UCT-C Taps to GigaVUE-FM for Volume-Based Licensing.

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 license distribution to individual nodes or devices becomes irrelevant for Gigamon accounting purpose. GigaVUE-FM tracks the total amount of data processed by the UCT-C and the overuse, if any.

- Volume-based has a service period of 1 month.
- The service period defines the duration during which the license is active.
- Total usage or overage is monitored.

To purchase licenses with the Volume-Based License (VBL) option, contact our Sales. Refer to Contact Sales.

Supported Platforms for UCT-C

The following table list the different platforms and their tested versions and Container Network Interface (CNI) that are qualified and supported for UCT-C.

Notes:

- As an end user, you must have an understanding and knowledge of your container services.
- UCT-C operates independently of any platform or CNI, and other CNIs are expected to function seamlessly. For any issues Contact Technical Support.

Platforms	Release Version	Tested Versions	CNI
Azure Kubernetes Service (AKS) **NOTE:** UCT-C is platform-independent and CNI agnostic solution and is expected to work with AKS in release versions 6.8 and 6.9.	v6.14	v1.33	The following listed CNIs are tested: • Calico • Flannel • Canal • Weave • Cilium • Open Virtual Network (OVN) **Note**: Support for Multus is currently unavailable.

Platforms	Release Version	Tested Versions	CNI
	v6.13	v1.33	

Platforms	Release Version	Tested Versions	CNI
	v6.12	v1.31	

Platforms	Release Version	Tested Versions	CNI
	v6.11	v1.31	

Platforms	Release Version	Tested Versions	CNI
	v6.10	v1.30	

Platforms	Release Version	Tested Versions	CNI
Amazon Elastic Kubernetes Service (EKS)	v6.14	v1.33	

Platforms	Release Version	Tested Versions	CNI
	v6.13	v1.33	

Platforms	Release Version	Tested Versions	CNI
	v6.12	v1.31	

Platforms	Release Version	Tested Versions	CNI
	v6.11	v1.31	

Platforms	Release Version	Tested Versions	CNI
	v6.10	v1.30	

Platforms	Release Version	Tested Versions	CNI
	v6.9	v1,26, v1.27, v1.28	
	v6.8	1,26, 1.27, 1.28	
Google Kubernetes Engine (GKE) **NOTE:** UCT-C is platform-independent and CNI agnostic solution and is expected to work with GKE in release versions 6.8 and 6.9.	v6.14	v1.33	
	v6.13	v1.33	
	v6.12	v1.31	
	v6.11	v1.30	
	v6.10	v1.29	
Rancher Kubernetes Engine **NOTE:** UCT-C is platform-independent and CNI agnostic solution and is expected to work with Rancher Kubernetes Engine in release version 6.11 and 6.12.	v6.10	v1.23 to v1.29	
	v6.9	v1.23 to v1.29	
	v6.8	v1.23 to v1.29	
Red Hat OpenShift Kubernetes Engine **NOTE:** The tested versions listed for Red Hat OpenShift (v4.13 and v4.14) correspond to OpenShift Container Platform (OCP) versions.	v6.14	v4.18	
	v6.13	v4.18	
	v6.12	v4.14	
	v6.11	v6.11	
	v6.10	v4.13 and v4.14	
	v6.9	v4.13 and v4.14	
	v6.8	v4.13 and v4.14	

Platforms	Release Version	Tested Versions	CNI
Native Kubernetes	v6.14	v1.33	
	v6.13	v1.33	
	v6.12	v1.29, v1.30	
	v6.11	v1.29, v1.30	
	v6.10	v1.29, v1.30	
	v6.9	v1.26 to v1.30	
	v6.8	v1.26 to v1.29	

NOTE: Previous versions, apart from those listed in the table above, are expected to function. For any issues Contact Technical Support.

UCT-C Resource Requirements

This section helps you learn how to initiate UCT Container and GigaVUE-FM deployment with the required resource requirements.

NOTE: You need a Load Balancer and an Ingress Controller to effectively manage ingress traffic. It ensures that applications within the cluster are accessible, secure, and performant.

For more details, refer to the following sections:

- Network Ports Requirements
- Compute Requirements
- Kernel and CPU Requirements for Universal Cloud Tap - Container

Network Ports Requirements

The tables below describe the Kubernetes network requirements for UCT Container to work efficiently.

Universal Cloud Tap - Container deployed inside Kubernetes worker node				
Direction	**Protocol**	**Port**	**Destination CIDR**	**Purpose**
Outbound	TCP	42042	Any IP address	Allows UCT-C to send statistical information to UCT-C Controller.
Outbound	UDP	VXLAN (default 4789)	Any IP address	Allows UCT-C to tunnel traffic to the GigaVUE V Series Node or

				other destination.
UCT-C Controller deployed inside Kubernetes worker node				
Direction	**Protocol**	**Port**	**Source CIDR**	**Purpose**
Inbound	TCP	8443 (configurable)	GigaVUE-FM IP	Allows GigaVUE-FM to communicate with UCT-C Controller.
Inbound	TCP	42042	Any IP address	Allows UCT-C Tap to communication with Controller.
Direction	**Protocol**	**Port**	**Destination CIDR**	**Purpose**
Outbound	TCP	5671	Any IP address	Allows UCT-C Controller to send statistics to GigaVUE-FM.
Outbound	TCP	443	GigaVUE-FM IP	Allows UCT-C Controller to communicate with GigaVUE-FM.

The following table describes the ports that you need to open on GigaVUE-FM:

Direction	Port	Purpose
Inbound	443	GigaVUE-FM REST service port.
Outbound	8443	Allows GigaVUE-FM to communicate with UCT-C Controller.
Inbound	5671	Allows UCT-C to send statistics to GigaVUE-FM through Rabbit-MQ port.

Compute Requirements

The following tables describes the minimum compute network requirements for UCT-C.

Compute Instances	vCPU	Memory	Disk Space
UCT-C Controller	1 vCPU	refer to the table below	—
UCT-C Tap	1 vCPU	1GB	—
GigaVUE V Series Node	4 vCPUs	8GB	20GB
GigaVUE V Series Proxy	1 vCPU	1GB	2GB
GigaVUE-FM	4 vCPUs	16GB	41GB

Compute Instances	Memory	Cluster Size
UCT-C Controller	256MB	less than 1000 pods
UCT-C Controller	512MB	2000 pods
UCT-C Controller	1GB	up to 5000 pods

Kernel and CPU Requirements for Universal Cloud Tap - Container

The supported kernel version requirements for different platforms are as follows:

- **Native Kubernetes** - 4.18 and above
- **GKE** - 5.15 (Ubuntu OS)
- **VMware Tanzu Photon**- 4.19 (Photon+ OS)

NOTE: UCT-C solution and its configuration would be successful only if deployed on worker nodes with Ubuntu 5.14 or later. If UCT-C Taps are deployed on worker nodes running GKE's Container Optimized OS (COS), the policy deployment from UI will fail with an "eBPF attachment error."

The supported CPU and RAM requirements for UCT-C and UCT-C Controller are as follows:

- **UCT-C TAP** - 1 vCPU and 1Gi
- **UCT-C Controller** - 1 vCPU and 1Gi

Deployment of UCT-C Solution

This section describes how you can deploy the UCT-C solution in a Kubernetes environment and configure in GigaVUE-FM.

For details, refer to the following sections:

- Launch GigaVUE-FM
- Deploy UCT-C Solution in Kubernetes
- Post Deployment
- Configure UCT-C Solution using GigaVUE-FM

Launch GigaVUE-FM

You can download the recent GigaVUE-FM image files from the Gigamon Customer Portal. After fetching the image, upload and launch GigaVUE-FM on your GigaVUE V Series Node supported cloud environment.

Contact Technical Support of Gigamon for assistance or for details on GigaVUE V Series configuration on the supported cloud environments, refer to GigaVUE Cloud Suite

NOTE: Ensure that GigaVUE-FM is up and ready before you deploy the UCT-C solution.

Deploy UCT-C Solution in Kubernetes

You can deploy UCT-C Solution in Kubernetes.

To deploy,

1. Implement external access to the Kubernetes environment (for example, ingress, external public IPs, load balancers) to allow communication between the UCT-C Controller and GigaVUE-FM. For details, refer to UCT-C Resource Requirements.
2. Verify if the firewall rules on Kubernetes nodes meet the defined requirements. For details, refer to Network Ports Requirements.
3. Untar the UCT-C image to access the Readme files.
   HELM charts are available as part of the UCT-C images.
4. Add the UCT-C images to a private Docker registry or ensure that the Docker Hub registry can pull the files. Y
5. Spin up or down the UCT-C instances based on traffic load.

You can deploy the UCT-C Controller and Taps in Kubernetes using the Helm Charts or Red Hat OpenShift Platform using OpenShift UI.

For details, refer to the following sections:

- Helm Charts
- Red Hat OpenShift Platform using OpenShift UI

Helm Charts

Helm Charts are available for every release build in the Gigamon software portal.

To get started,

1. Download the respective UCT-C release build from the repository.
2. Untar the **gigamon-gigavue-uctc-helm-6.14.tgz** file.
3. Navigate to the Helm folder.
4. Update the fields mentioned in Deploy using Helm Chart before deployment.

Notes:

- Contact Technical Support or Contact Sales for information on downloading the respective UCT-C build from the Gigamon software portal.
- Support for two Helm Charts is deprecated from software version 6.7.00.

Using Helm with an External values.yaml File

To deploy the chart in an external file, follow these steps:

1. Download the original Helm chart from Gigamon software portal locally or to a remote Helm chart repository.
2. Untar/zip the chart.
3. Copy the original **values.yaml** file to a new file named '**myvalues.yaml**'.
4. Edit the **myvalues.yaml** file with the necessary changes as mentioned in Deploy using Helm Chart.

Use the below deployment command to install the chart:

```
helm install uctc -f myvalues.yaml gigamon-gigavue-uctc-helm-6.14.tgz -n uct-ns
```

Deploy using Helm Chart

You can deploy UCT-C Controller and TAPs using Helm Chart.

To deploy,

1. Use the command below to Unzip and Untar the .tgz file:

   ```
   gunzip <name of the UCT-C .tgz file>
   tar -xvf <name of the UCT-C .tar file>
   ```

2. After extracting the tar file, navigate to the Helm folder in the newly created **uctc-<image version>-<build number>** folder.
3. Update the necessary configuration fields (e.g., imagePullSecrets, namespace, GigaVUE-FM IP, external load balancer IP, and Kubernetes API URL) in the values.yaml file.

   ```
   imagePullSecrets: [{name: secret}]
   namespace: uctc
   ```

   If only an IPv4 address is provided and IPv6 is not configured, GigaVUE-FM uses the IPv4 address exclusively for communication.

   ```
   fm_ip: "<FM IPv4>"
   ```

   If only an IPv6 address is provided and fm_ip is not configured, GigaVUE-FM uses the IPv6 address for communication.

   ```
   fm_ipv6: "<FM IPv6>"
   ```

   If both IPv4 and IPv6 are provided, **UCTC_CNTLR_FM_IP_CONFIG** is used to choose the preferred IP stack.

```
ext_load_balancer: "<FM IPv4 or FM IPv6 or FM IPv4,FM IPv6(both with comma-separation notation>"
Refer to the below examples:
# example1: 192.168.0.10 (IPv4)
# example2: 2001:db8:abcd:ef01::5 (IPv6)
# example3: 192.168.0.10,2001:db8:abcd:ef01::5 (IPv4,IPv6)
k8s_cluster_url: "<url>"
# example: https://10.10.10.12:6443
```

NOTE: Helm Chart supports OpenShift Routes for external communication within OpenShift clusters. During the upgrade, delete any manually created UCT-C Route and use the Helm-managed Route instead. Existing UCT-C Routes are not supported post-upgrade.

4. If you provide two IPs, IPv4 and IPv6 in the fm_ip argument, use the following configurations:

```
# values: <IPv4 | IPv6>
uctc_tap_ip_config: "IPv4"
# values: <true | false>
uctc_tap_fallback_config: "false"
```

IP CONFIG option allows the user to provide the preferred IP version. If the user does not provide any value, the default value IPv4 is used.

When the preferred IP version fails to connect (example: IPv6), the **FALLBACK CONFIG** is used to connect to the other available IP version (example: IPv4). The default value, True, is used to consider the Fallback mechanism.

Notes:

- Fallback configuration is used during the node registration phase only.
- Controller FM IP and FALLBACK configurations is used only if you provide both IPv4 and IPv6.
- Default values is used if you do not provide any options.

5. Edit the following **volumeMounts** as per your container Runtime:

```
crisocketvolume:
mountPath: /var/run/containerd/containerd.sock
name: socket
```

The socket location for commonly used CRIs are as follows:

```
docker - /var/run/docker.sock
containerd - /var/run/containerd/containerd.sock
cri-o - /var/run/crio/crio.sock
```

6. Run the below command in the location where the UCT-C folder is present.

```
helm install uctc /uctc -n <Namespace>
```

> **NOTE:** You can skip steps 1-5 and use the below command to directly deploy UCT-C Controller and TAPs using the Helm Chart.
> Example command:
> helm install uctc -n uctc ./uctc \
> --set namespace=uctc \
> --set serviceAccount.name=test \
> --set imagePullSecrets[0].name=gigamon \
> --set ingress.ext_load_balancer=<EXTERNAL_LB_IP> \
> --set uctcController.fm_ip=<FM_IP> \
> --set uctcController.k8s_cluster_url=https://<K8S_API_URL>:6443 \
> --set uctcController.uctc_cntlr_fm_ip_config=IPv4 \
> --set uctcTap.uctc_tap_ip_config=IPv4 \
> --set uctcTap.cri_socket_path=/run/containerd/containerd.sock
> Replace placeholder values such as <FM_IP>, <EXTERNAL_LB_IP>, and <K8S_API_URL> with environment-specific details. To include additional parameters defined in the values.yaml file, append more --set options to the command as needed.

To include additional parameters defined in the values.yaml file, append more --set options to the command as needed.

Validate UCT-C Deployment

To validate the UCT-C deployment and check for any failures, set the validation value to "True" in values.yaml file. Two pods, **uctc-prevalidator-pod** and **uctc-postvalidator-pod** are deployed to perform checks and ensure certain conditions are met for a successful deployment.

- If all the checks pass, the validator pods clean up automatically, and UCT-C is deployed successfully.
- If any check fails, the respective validator pod (either **uctc-prevalidator-pod** or **uctc-postvalidator-pod**) remain in an error state, and the Helm installation fails.

Perform the following steps if the installation fails due to a validation error:

1. **Check Logs:** To review the logs of the failure pod and identify the issue, use the following command and specify the corresponding failed pod name.

```
kubectl logs < uctc-prevalidator-pod | uctc-postvalidator-pod > -n uctc
```

2. **Delete the Helm Release:** Use the following command to delete the failed Helm release.

```
helm uninstall my-release -n uctc
```

3. **Delete the Error Pod:** After you identify the cause of failure, you can delete the failed validator pod (uctc-prevalidator-pod or uctc-postvalidator-pod depending on which pod fails) and re-run the Helm installation. Use the command below to delete the failed validator pod.

   ```
   kubectl delete pod <uctc-prevalidator-pod | uctc-postvalidator> -n uctc
   ```

4. Run the command below in the location where the UCT-C folder is present.

   ```
   helm install uctc /uctc -n <Namespace>
   ```

   NOTE: If intermittent connectivity issues persist between GigaVUE-FM and the cluster, set 'validation' to false before installing the Helm chart to avoid false negative failures.

The following table lists the configurable parameters and their default values defined in the values.yaml file:

UCT-C Controller configuration

Parameter	Description	Default Value
uctcController.image.repository	UCT-C Controller Docker image repository.	gigamon/gigamon-gigavue-uctc-cntlr
uctcController.image.tag	UCT-C Controller Docker image tag.	XXX_IMAGE_TAG_XXX
uctcController.nameOverride	This value overrides the default resource's name generated by the chart's templates.	uctc-cntlr
uctcController.fullnameOverride	The provided name combines with the default resource's name.	
uctcController.podAnnotations	Annotations are added based on the user requirements.	
uctcController.service.name	Name of the UCT-C Controller Service that you need to create.	uctc-cntlr-service
uctcController.service.type	Type of the service that you need to create.	ClusterIP
uctcController.fm_ip	IPv4 address of the GigaVUE-FM. • If only IPv4 is provided and fm_ipv6 is not specified, GigaVUE-FM by default uses IPv4 for communication. • If both fm_ipv4 and fm_ipv6 are provided, the preferred IP stack is selected based on the configuration specified in uctc_cntlr_fm_ip_config.	
uctcController.fm_ipv6	IPv6 address of the GigaVUE-FM.	

Parameter	Description	Default Value
	• If only IPv6 is provided and fm_ip is not specified, UCT-C Controller uses IPv6 for communication. • If both fm_ipv4 and fm_ipv6 are provided, the preferred IP stack is determined based on the uctc_cntlr_fm_ip_config setting.	
uctcController.uctc_svc_rest_port	Port at which UCT-C Controller listens the GigaVUE-FM request. **NOTE:** Set this value to 443 if OpenShift Route is enabled.	8443
uctcController.fm_svc_rest_port	Port at which GigaVUE-FM listens the UCT-C Controller registration message.	443
uctcController.k8s_cluster_url	K8S cluster end-point (typically, master nodes with the default port of 6443).	
uctcController.uctc_cntlr_fm_ip_config	User's preferred IP stack for communication between the UCT-C Controller and a dual-stack GigaVUE-FM.	IPv4
uctcController.uctc_cntlr_fallback_config	If enabled as 'True,' the uctc_cntlr_fm_ip_config utilizes the alternative IP stack if the preferred one is unavailable.	True
uctcController.fm_fqdn	Specifies the GigaVUE-FM HA FQDN value used by the UCT-C Controller for communication with the GigaVUE-FM HA group. When this parameter is set, the controller uses the IP addresses resolved from the FQDN. **Example fm_fqdn**: "fm.gigamon.com"	
uctcController.fmha_ip_list	Specifies a comma-separated list of GigaVUE-FM HA IP addresses used by the UCT-C Controller for communication with the GigaVUE-FM HA group. If fm_fqdn is not set, the controller uses the IP addresses provided in fmha_ip_list by default. **Example fmha_ip_list**: "10.0.0.1,10.0.0.2,10.0.0.3"	
uctcController.resources.limits.cpu	Expressed in CPU units, it represents the maximum processing power that the UCT-C controller pod is allowed to use.	1
uctcController.resources.limits.memory	Expressed in Gibibytes, it represents the	1Gi

Parameter	Description	Default Value
	maximum amount of RAM that the UCT-C controller pod is allowed to consume.	
uctcController.resources.requests.cpu	Expressed in CPU units, it represents the minimum processing power that the UCT-C controller pod needs.	1
uctcController.resources.requests.memory	Expressed in Gibibytes, it represents the minimum amount of RAM that the UCT-C controller pod needs.	1Gi
uctcController.nodeSelector	Node labels specified to target specific nodes in the Kubernetes cluster where the UCT-C Controller pod is scheduled.	
uctcController.tolerations	Toleration's specified to tolerate specific taints on nodes.	
uctcController.affinity	Affinity rules specified to place UCT-C Controller pod on nodes based on complex rules.	

NOTE: The controller includes a service port that is now provided to FM. By default, the controller pod continues to listen on port 8443, as defined by the command parameter. To expose the service on a different port, such as 443, you can configure this in the Service YAML. Kubernetes manages the redirection, allowing external access through the specified port (443), while the controller internally continues to operate on port 8443.

Example: Sample Configuration in the Service YAML:

```
namespace: uct
spec:
 ports:
 port: 443
 protocol: TCP
 name: uct-rest
 targetPort: 8443
```

UCT-C Tap configuration

Parameter	Description	Default Value
uctcTap.image.repository	UCT-C Tap Docker image repository.	gigamon/gigamon-gigavue-uctc-tap
uctcTap.image.tag	UCT-C Tap Docker image tag.	XXX_IMAGE_TAG_XXX
uctcTap.nameOverride	This value overrides the default resource's	

Parameter	Description	Default Value
	name generated by the chart's templates.	
uctcTap.fullnameOverride	This value combines with the default resource's name.	
uctcTap.podAnnotations	Annotations to add to the pod.	
uctcTap.podSecurityContext	Security context to add to the pod.	
uctcTap.uctc_tap_ip_config	User's preferred IP stack for communication between the UCT-C Controller and UCT-C Tap.	IPv4
uctcTap.uctc_tap_fallback_config	If enabled as 'True,' the uctc_tap_ip_config, utilizes the alternative IP stack if the preferred one is unavailable.	True
uctcTap.uctc_policy_resp_sz	This configuration sets the policy response size in terms of the number of pods. If the configuration includes 64 rules per pod, it is recommended to set the pod count to 10. **Valid range**: <POLICY_RESP_SZ: 1 pod (minimum) to 50 pods (maximum)>	25
uctcTap.resources.limits.cpu	Expressed in CPU units, it represents the maximum processing power that the UCT-C Tap pod is allowed to use.	1
uctcTap.resources.limits.memory	Expressed in mebibytes, it represents the maximum amount of RAM that the UCT-C Tap pod is allowed to consume.	1Gi
uctcTap.resources.requests.cpu	Expressed in CPU units, it represents the minimum processing power that the UCT-C Tap pod needs.	1
uctcTap.resources.requests.memory	Expressed in mebibytes, it represents the minimum amount of RAM that the UCT-C Tap pod needs.	1Gi
uctcTap.cri_socket_path	Key-in the container run-time specific socket path for, for example for cri-o -> "/run/crio/crio.sock" containerd -> /run/containerd/containerd.sock and docker -> /var/run/docker.sock.	
uctcTap.nodeSelector	Node labels specified to target specific nodes in the Kubernetes cluster where the UCT-C Tap pod is scheduled.	
uctcTap.tolerations	Toleration’s specified to tolerate specific taints on nodes.	
uctcTap.affinity	Affinity rules specified to place UCT-C Tap pod on nodes based on complex rules.	

Common Configuration

Parameter	Description	Default Value
namespace	Namespace to deploy the UCT-C components.	uct
serviceAccount.create	Specifies creation of a service account.	FALSE
serviceAccount.annotations	Annotations to add to the service account.	
serviceAccount.name	Name of the service account if uctcTap.serviceAccount.create is set to true.	gigamon
imagePullSecrets	List of image pull secrets for private registries.	gigamon (customize as needed)
openShift.enabled	Enables OpenShift-specific deployment settings when set to true.	FALSE
openShift.route	Defines configuration options for OpenShift Route integration. If using openshift route please use port 443 for uctc_svc_rest_port.	
openShift.route.create	Creates a new OpenShift Route if set to true.	FALSE
openShift.route.name	Specifies the name of the route to create or use.	uct-route
openShift.route.baseDomain	Sets the base domain for the route. Use **oc get dns cluster -o=jsonpath='{.spec.baseDomain}'** to get the base domain.	
ingress.ext_load_balancer	External loadbalancer IP or DNS through which GigaVUE-FM communicates with the UCT-C Controller. **NOTE:** You should leave this field empty if OpenShift Route is enabled.	
ingress.enabled	Set this parameter to **true** when using an ingress for external access. **NOTE:** For an OpenShift cluster, set either openShift.route.enabled or ingress.enabled to true—not both.	TRUE
ingress.create	This parameter determines whether Helm should create a new Ingress resource. • Set to "true" if you want Helm to generate and manage the Ingress resource.	TRUE

Parameter	Description	Default Value
	• Set to "false" if you already have an existing Ingress resource or prefer to manage it separately outside this Helm chart.	
ingress.annotations	Annotations added to the ingress.	
ingress.annotations.kubernetes.io/ingress.class	Class name of the ingress controller to be used.	nginx
debugmode	Specified in the hex form of 0x00[aaaa][b][c] where, • aaaa is the number of pcap messages to maintain before rollover • b can have the value 0 or 1 where 0=do not create pcap or 1=create pcap • c can have the value between 1 to 4 where 1=fatal, 2=error, 3=info, 4=debug.	0x0A000003
securityContextConstraints.create	If the platform is OpenShift, set securityContextConstraints.create to true. This creates a custom Security Context Constraint (SCC) with the required permissions for the UCT-C solution. When using privileged ports, ensure to launch the controller with root privileges.	FALSE
securityContextConstraints.name	Name of the Security context constraints (SCC) that you create.	gigamon
validation	Setting this to true deploys a pre-validator pod before and a post-validator pod after the deployment of the UCT-C solution.	TRUE

You should specify each parameter using the --set key=value argument with the helm install command. You can customize the Helm chart by modifying the values in the values.yaml file or by using the --set flag with the helm install command.

Red Hat OpenShift Platform using OpenShift UI

You can deploy the UCT-C Controller and Taps in the Red Hat OpenShift Platform using Helm Charts.

Refer to the following sections:

- Prerequisites
- Deployment of UCT-C Controller and Taps

Prerequisites

- To deploy, you should have Developer access in Red Hat OpenShift Platform.
- To validate the deployment, you should have Administrator access.

Deployment of UCT-C Controller and Taps

To deploy UCT-C Controller and Taps, follow these steps:

1. Using your Red Hat login credentials. log in to the Red Hat OpenShift online platform.
2. Select the drop-down on the top of the page and switch to Developer access.
3. Navigate to the **Helm** section, and select **Create > Helm Release**.
   The Helm Charts screen appears.
4. From the **All items** search menu, browse and select Gigamon.
5. On the Gigamon-UCT-C landing page, select **Create**.
   The Create **Helm Release** page appears.

   > **NOTE:** The **README** content on the Gigamon-UCT-C landing page provides information on how to deploy the UCT-C Controller and Tap on a Kubernetes cluster using Helm Chart.

6. Specify the Helm Release name to create Helm Release
7. Select the appropriate release version from the drop-down menu. By default, the latest uploaded version of the release is displayed.
8. Select either Form view or YAML view for configuration and specify the created secret name in imagePullSecrets field.
9. In the **uctcTap** section:
   a. Specify the socket location details in resources - crisocketvolume filed.

   > **NOTE:** The socket location for commonly used CRIs are as follows:
   > **docker** - /var/run/docker.sock
   > **containerd** - /var/run/containerd/containerd.sock
   > **cri-o** - /var/run/crio/crio.sock

   b. Specify the namespace.
   c. Specify the following details in the ingress section:
      i. **enabled** - Select the check box to enable.
      ii. **annotations** - Specify the annotations details (kubernetes.io/ingress.class and nginx.ingress.kubernetes.io/backend-protocol).
   d. Enable the Create option and specify the serviceAccount name.

10. In the **ingress** section, specify the **ext_load_balancer** details. Provide the external load balancer IP address or DNS name to enable communication between GigaVUE-FM and the UCT-C Controller running in Kubernetes.
11. In the **uctcController** section:
    a. Specify the port value.
    b. In the certs field, specify details in **k8s_cluster_url** — Kubernetes cluster endpoint that GigaVUE-FM uses to access the control plane.
    Example: https://<kubernetesapiserverurl>:6443
    c. Specify the service label name. For example: uctc-cntrl-service.
    d. Specify the repository and pullPolicy details in the image field.
    e. Update the namespace and fm_ip details.
12. Select **Create** to deploy the UCT-C solution.
13. To validate the deployment, switch to **Administrator** view and navigate to:
    a. **DaemonSets** option to validate the UCT-C-Tap deployment.
    b. **Deployment** option to validate the UCT-C-Controller deployment.

Post Deployment

After you deploy UCT-C Solution in Kubernetes, perform the following for post deployment verification checks:

- Verify UCT-C deployment using CLI
- Verify UCT-C deployment using GigaVUE-FM

Verify UCT-C deployment using CLI

Use the following CLI commands to ensure that the UCT-C components are up and running properly within the specified namespace.

1. Check the Controller Deployment:
   ```
   kubectl get deployment -n <namespace>
   ```
   This command verifies that the UCT-C controller deployment exists and is running.
2. Check the TAP DaemonSet Status:
   ```
   kubectl get daemonset -n <namespace>
   ```
   This command confirms that TAP DaemonSet is running.
3. Verify if the Controller and TAP Pods are Up and Running:
   ```
   kubectl get pods -n <namespace>
   ```
   This command displays the current status of the Controller and TAP pods.

4. Confirm the Controller Service is Present and Accessible:

   ```
   kubectl get svc -n <namespace>
   ```

   This command confirms that the controller service is correctly deployed and accessible within the cluster.

5. Check the Ingress Resource (if applicable):

   ```
   kubectl get ingress -n <namespace>
   ```

   If using the Ingress resource for routing external traffic, use this command to verify the Ingress configuration for the controller.

6. Check the Route Resource (if applicable):

   ```
   kubectl get route -n <namespace>
   ```

   If using a Route resource to direct external traffic to the appropriate service in the cluster, use this command to confirm the route configuration.

7. Verify Helm Chart Deployment (if using Helm):

   ```
   helm list
   ```

   This command lists the Helm releases deployed in your cluster.

Verify UCT-C deployment using GigaVUE-FM

You can verify the following interactions between UCT-C and GigaVUE-FM:

1. Register UCT-C Controller and TAP
2. Check UCT-C Controller and TAP Status
3. Check UCT-C Controller Connectivity
4. Verify the Cluster Inventory

Register UCT-C Controller and TAP

When a UCT-C Controller and TAP start in the Kubernetes environment, they register directly with GigaVUE-FM. Check the network requirements for a successful registration. For details, refer to Network Ports Requirements.

UCT-C supports IPv4 and IPv6 protocols. For details, refer to Deploy UCT-C Solution in Kubernetes.

- When UCT-C Controller and TAP terminate normally, it sends the deregistration message to GigaVUE-FM.
- If UCT-C Controller and TAP go down abnormally and GigaVUE-FM fails to receive a couple of heartbeats, it will get disconnected.

Check UCT-C Controller and TAP Status

GigaVUE-FM marks the heartbeat status of UCT-C Controller and TAP as **Connected** when it registers.

- After successful registration, UCT-C Controller and TAP send heartbeats to GigaVUE-FM every 30 seconds.
- GigaVUE-FM scans the last received heartbeat of the registered UCT-C Controller and TAP pods and marks the heartbeat status periodically (1 minute).

  Scenarios for heartbeat status:

- **Pending**: Missed 2 consecutive heartbeats
- **Disconnected**: Missed 3 consecutive heartbeats
- GigaVUE-FM automatically purges disconnected or terminated UCT-C Controllers and TAPs after a retention period of 7 days.

NOTE: GigaVUE-FM generates an alarm for the disconnected UCT-C when three consecutive heartbeats are missed. For details, refer to "Alarms" topic in the *GigaVUE Administration Guide* .

Check UCT-C Controller Connectivity

After successful registration of UCT-C Controller, GigaVUE-FM periodically checks connectivity to Controller.

Connectivity status:

- **Reachable:** If GigaVUE-FM can connect with the UCT-C Controller
- **Unreachable:** If GigaVUE-FM cannot connect with the UCT-C Controller

NOTE: You should ensure that the UCT-C Controller connectivity is Reachable before configuring it. If the connectivity shown for a UCT-C controller in a cluster is not reachable, the deployment for that cluster does not go through.

Verify the Cluster Inventory

You can navigate to **Inventory > CONTAINER > Universal Cloud Tap - Container** to verify the UCT-C cluster inventory in GigaVUE-FM. In the Clusters, Nodes, and Pods sections, you can:

- Verify that all clusters are listed, operational, and running.
- Check the cluster's health status, connectivity, and resource usage.

Verify UCT-C deployment using CLI

You can verify the UCT-C deployment using the following CLI commands to ensure that the UCT-C components are up and running as expected within the specified namespace.

1. Check for the Controller Deployment:

   ```
   kubectl get deployment -n <namespace>
   ```

   This command verifies that the UCT-C controller deployment exists and is running.

2. Verify the TAP DaemonSet Status:

   ```
   kubectl get daemonset -n <namespace>
   ```

   Use this command to check the status of the TAP DaemonSet, ensuring that it is properly deployed.

3. Verify if the Controller and TAP Pods are Up and Running:

   ```
   kubectl get pods -n <namespace>
   ```

   This command checks that the controller and TAP pods are running and available.

4. Confirm the Controller Service is Present and Accessible:

   ```
   kubectl get svc -n <namespace>
   ```

   Run this command to confirm that the controller service is correctly deployed and accessible within the cluster.

5. Check the Ingress Resource (if applicable):

   ```
   kubectl get ingress -n <namespace>
   ```

   If using the Ingress resource for routing external traffic, use this command to verify the Ingress configuration for the controller.

6. Check the Route Resource (if applicable):

   ```
   kubectl get route -n <namespace>
   ```

   If you are using a Route resource to direct external traffic to the appropriate service in the cluster, use this command to confirm the route configuration.

7. Verify Helm Chart Deployment (if using Helm):

   ```
   helm list
   ```

   This command lists the Helm releases deployed in your cluster.

Verify UCT-C deployment using GigaVUE-FM

You can verify the following interactions between UCT-C and GigaVUE-FM:

- Register UCT-C Controller and TAP
- Check UCT-C Controller and TAP Status
- Check UCT-C Controller Connectivity
- Verify the Cluster Inventory

Register UCT-C Controller and TAP

When a UCT-C Controller and TAP come up in the Kubernetes environment, it registers itself with GigaVUE-FM.

Check the network requirements for the registration to be successful. For more information, refer to Network Ports Requirements.

UCT-C supports IPv4 and IPv6 protocols. For more information, refer to Deploy UCT-C Solution in Kubernetes.

When UCT-C Controller and TAP is terminated normally, it sends the deregistration message to GigaVUE-FM. If UCT-C Controller and TAP goes down abnormally and GigaVUE-FM fails to receive a couple of heartbeats, it will get disconnected.

Check UCT-C Controller and TAP Status

GigaVUE-FM marks the heartbeat status of UCT-C Controller and TAP as **Connected** when it gets registered. After successful registration, UCT-C Controller and TAP sends heartbeats to GigaVUE-FM every 30 seconds. GigaVUE-FM scans the last received heartbeat of the registered UCT-C Controller and TAP pods and marks the heartbeat status periodically (1 minute). The following are the various scenarios where the heartbeat status changes:

- If 2 consecutive heartbeats are missed, GigaVUE-FM marks the status as **Pending**.
- If 3 consecutive heartbeats are missed, GigaVUE-FM marks the status as **Disconnected**.
- GigaVUE-FM purges disconnected or terminated UCT-C Controllers and TAPs after 7 days.

NOTE: GigaVUE-FM generates an alarm for the disconnected UCT-C when three consecutive heartbeats are missed. Refer to "Alarms" topic in the *GigaVUE Administration Guide* for detailed information on Alarms.

Check UCT-C Controller Connectivity

After successful registration of UCT-C Controller, GigaVUE-FM periodically checks connectivity to Controller. The following are the various scenarios where the UCT-C Controller connectivity changes:

- If GigaVUE-FM can connect with the UCT-C Controller, the connectivity status will be marked as **Reachable**.
- If GigaVUE-FM cannot connect with the UCT-C Controller, the connectivity status will be marked as **Unreachable**.

NOTE: You should ensure that the UCT-C Controller connectivity is Reachable before doing any configurations. If the connectivity shown for a UCT-C controller in a cluster is not reachable, the deployment for that cluster will not go through.

Verify the Cluster Inventory

You can verify the UCT-C cluster inventory in GigaVUE-FM by navigating to the **Inventory > CONTAINER > Universal Cloud Tap - Container**. In the Clusters, Nodes, and Pods sections you can verify that all clusters are listed, operational, and running. You can also check the cluster's health status, connectivity and resource usage.

Configure UCT-C Solution using GigaVUE-FM

This section describes how to configure UCT-C through GigaVUE-FM.

Refer to the following section for details.

- Universal Cloud Tap - Container Inventory
- Create Monitoring Domain
- Create Source Selectors
- Create Tunnel Specifications
- Configure Traffic Policy
- View Policy Configurations
- View Traffic Policy Statistics

Universal Cloud Tap - Container Inventory

On the left pane of GigaVUE-FM, go to **Inventory > CONTAINER > Universal Cloud Tap - Container**.

You can view the following tabs on the Universal Cloud Tap - Container launch page:

Tabs	Description
Monitoring Domains	Displays the Monitoring Domain details and the connectivity status from GigaVUE-FM to Cluster. The count of reachable and unreachable clusters per Monitoring Domain.
Clusters	Displays Kubernetes Clusters, along with UCT-C Controller information and the total number of nodes per Cluster. Also displays GigaVUE-FM to Controller connectivity and Heartbeats status. Heartbeat status is from UCT-C Controller to GigaVUE-FM. **Note**: • You can delete a cluster that is not associated to any Monitoring Domain from the Clusters page. Ensure that you delete the cluster only after stopping the UCT-C Controller and the UCT-C TAP. • UCT-C solution supports a maximum of 64 clusters, with a default maximum of 50. Refer to Configure UCT-C Settings.
Nodes	Displays the Nodes from all Kubernetes Clusters along with UCT-C TAP information and Total Pods per Node. UCT-C TAP status should be connected

Tabs	Description
	for deployments for respective Worker Nodes to go through. If the UCT-C TAP status for a Worker Node is not shown as Connected, the deployment for that Worker Node will not go through. **NOTE:** A maximum of 500 nodes are supported per cluster, with qualified support available for up to 250 nodes. For any issues, Contact Technical Support.
Pods	Displays the list of Pods from all Kubernetes Clusters. For each Pod, all metadata - Pod Name, Labels, IPs, Namespace, Service Name, Service IPs, Node Name, Containers, and Host Network information is displayed. **Service Mesh Mode** – Indicates whether the pod is Service Mesh enabled and shows the mode. Currently, only Sidecar mode is supported. **Service Ports** – Lists the service ports associated with each pod, in addition to the Service IPs. **NOTE:** A maximum of 25K pods are qualified across single or multiple clusters. Pod counts beyond this limit are expected to function. For any issues, Contact Technical Support.
Settings	Displays the general settings which include disconnected UCT-C Controller or TAP Purge Interval days, and the maximum number of Clusters allowed in GigaVUE-FM.

To view and filter the list of Monitoring Domains, Cluster, and Node details, perform one of the following:

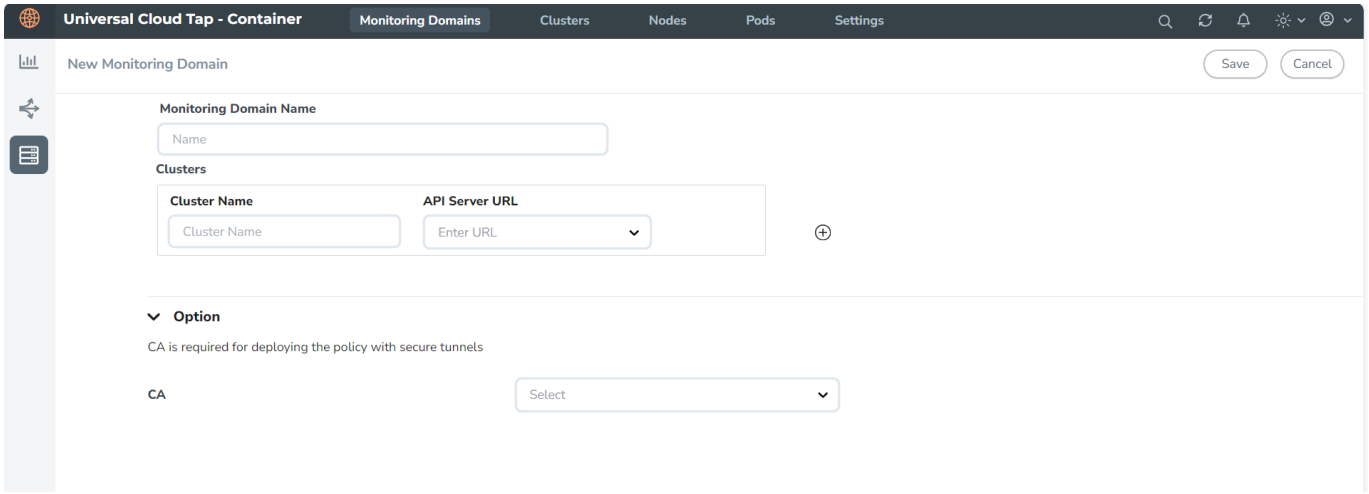
- Select the  filter button on the left side of any of the above-listed tabs.
- Create a new Monitoring Domain, edit, and delete the existing Monitoring Domains.

On Clusters, Nodes, and Pods screens, you can click the **Filter** button on the right side to filter the details on that particular screen.

Create Monitoring Domain

To create a monitoring domain in GigaVUE-FM:

1. Go to **Inventory > CONTAINER > Universal Cloud Tap - Container > Monitoring Domains**.
2. In the **Monitoring Domains** page, select **New**.
   The **New Monitoring Domain** wizard appears.



3. Enter the name of the Monitoring Domain and the Cluster Name.
4. Enter or select the URL of the API server.
5. Select the required name for CA.

   NOTE: CA is required to deploy the policy with Secure Tunnels.

6. Select ⊕ to add another cluster or select ⊖ to remove an existing cluster.
7. Select **Save**.

You can view the Monitoring Domain created in the list view. The list view shows the following information for UCT-C and controllers:

- **Monitoring Domain:** Shows the list of Monitoring Domains created.
- **Cluster:** Displays the status of GigaVUE-FM to UCT-C Controller connectivity.

  You can click the number link next to ✓ (connected) or the ⬡ (disconnected) icons to view the cluster details for the selected Monitoring Domain.

Use the following buttons to manage your Monitoring Domain:

- **New**: Use to create a new Monitoring Domain.
- **Actions**: Use to edit or delete the Monitoring Domain(s).
- **Refresh Inventory**: Triggers Inventory Refresh on all Clusters in the Monitoring Domain.

NOTE: IP forwarding should be enabled on all worker nodes in the cluster.

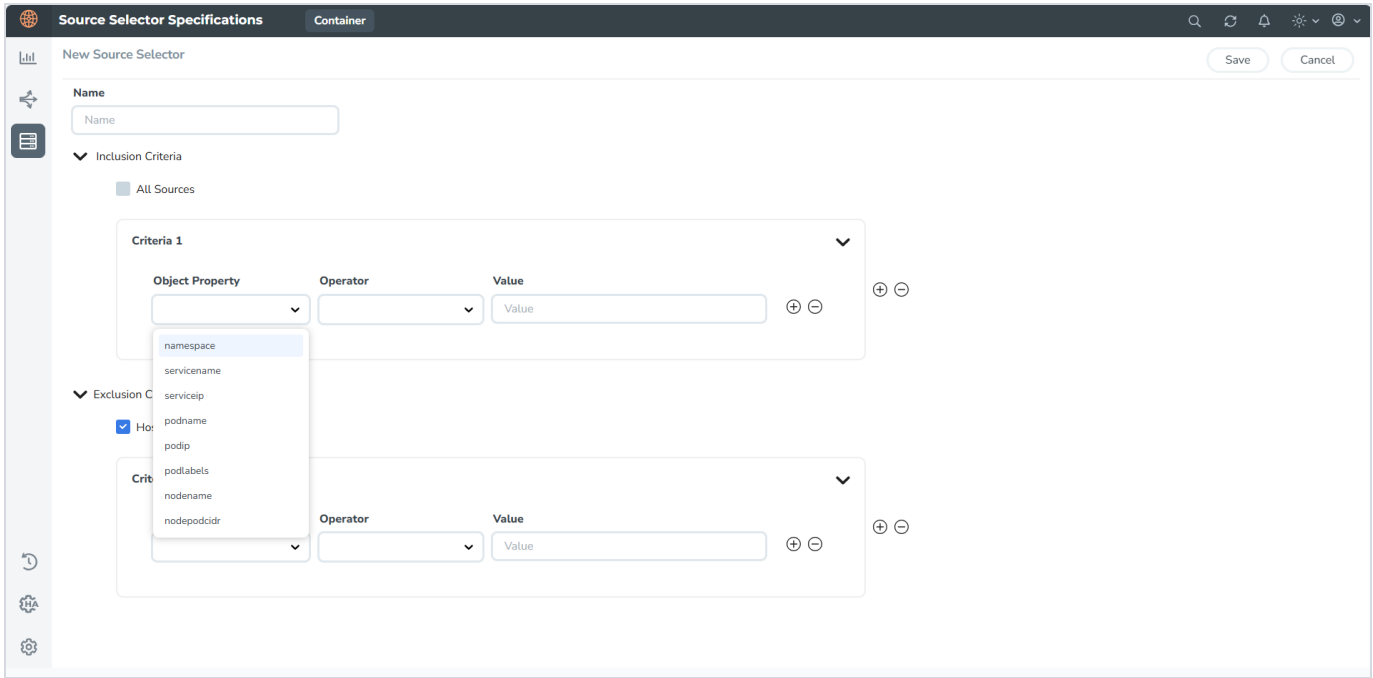
Create Source Selectors

When setting up a traffic flow, it is important to define the selection criteria for the sources of traffic. You should configure the sources of the traffic that you want to monitor.

DefaultExclusion: It is a default source selector applied to all policies. You can modify but not delete it. After modifying the DefaultExclusion source selector, you need to redeploy policies for the changes to take effect. DefaultExclusion appears by default on the **Source Selector Specifications** page.

- To exclude the pods from monitoring, you can add criteria to DefaultExclusion. From the drop-down list, select any of the following Object Property to exclude them from the monitoring, and provide the value for the property selected in the value field:
  - servicename
  - serviceip
  - podname
  - podip
  - podlabels
  - nodename
  - namespace
  - nodepodcidr

  By default, pods in kube-system namespace, and metallb-system namespace are excluded from monitoring.
- You can add criteria to DefaultExclusion to exclude nodenames where UCT-C TAP is not launched. If Master node(s) does not have UCT-C TAP, add master node names to DefaultExclusion.
- During the upgrade from version less than 6.10, GigaVUE-FM automatically removes the Host Network criteria from the DefaultExclusion Source Selector and adds the Host Network Enabled **Exclusion Criteria** by default to all the existing policies.



Notes:

- It is recommended to use **istioctl** to install Istio. This ensures all required symbol tables are installed and the solution functions correctly. Installing Istio through alternate methods, such as Red Hat OpenShift, may cause the Service Mesh Clear Text solution to not work as expected.
- In a Service Mesh clear-text policy, L4 filter rules work only when Istio-enabled worker pods and services expose a single port.

To configure the Source Selectors,

1. Go to **Inventory** > **Resources> Source Selectors> Container**.
2. On the **Source Selector Specifications** page, navigate to the **Container** tab.
3. On the **Source Selector Specifications** page for Containers, select **Create**. The **New Source Selector** wizard appears.
4. Enter the name of the source.
5. In the **Inclusion Criteria**, select from the following options:
   a. **All Sources:** Select this option to acquire traffic from all namespaces and pods within the selected cluster(s). The traffic volume may be large, depending on the cluster(s) size.
   b. **Criteria1:** Enter the following options:
      i. Select an object property to filter the traffic source.
      ii. Select the operator and enter the values for the filter (values are case-sensitive).
6. In the **Exclusion Criteria**, select or enter the following options:
   a. Select an object property to filter the traffic source.
   b. Select the operator and enter the values for the filter (values are case-sensitive).
   c. Select the Host Network Enabled to view the configuration in your policies. The UCT Container TAP introduces support for tapping Host Network Enabled pods. For details, refer to the Exclusion Criteria.
7. Select **Save**.

Notes: You can create multiple criteria. Within each criteria, you can configure multiple objects.

- If you configure multiple objects in a criteria, the traffic is filtered only if all the object rules are true (AND condition).
- If you configure multiple criteria, then the traffic is filtered even if one of the criteria is true (OR condition).

Exclusion Criteria

Host Network Enabled - The UCT-C introduces support for tapping Host Network Enabled pods. By default, this check box is selected indicating that you are excluding the host network enabled pods.

When you want to monitor the pod, clear the **Host Network Enabled** check box. A warning message appears and requires your confirmation to proceed with tapping pods with Host Network Enabled.

Service Mesh Enabled – UCT-C supports tapping pods that are part of a Service Mesh. By default, Service Mesh-enabled pods are excluded from Precryption policies.

- To monitor Clear text traffic from Service Mesh-enabled pods, select the Service Mesh Clear Text policy type.
- In Mirroring policies page, you can choose whether to include or exclude Service Mesh-enabled pods based on your monitoring needs.

Notes:

- Worker Node must have cgroup version 2 to support the Host Network Enabled feature.
- If the Worker Node has cgroup version 1, the policy deployment status for pod shows an error message.
- When tapping Host Network enabled pods, tapped traffic is sent to user space for tunneling. It uses performance buffers, requiring more memory. To accommodate this, increase the memory request/limit to at least 1GB for UCT-C taps.

Identify the cgroup Version on the Worker Node

Use one of the following options to verify which cgroup version your distribution uses:

- Run the **stat -fc %T /sys/fs/cgroup/** command on the worker node:
  - For cgroup v2, the output is**cgroup2fs**.
  - For cgroup v1, the output is **tmpfs**.
- Check if **/sys/fs/cgroup/cgroup.controllers** is present, then it is cgroup v2.

Identify the cgroup Version for Worker Pod

To check which cgroup version your worker pod uses:

1. Log in to the worker pod.
2. Open the file **/proc/$$/cgroup** and check if the file contains **net_cl**.
   The file with **net_cls** represent cgroup v1. The file without **net_cls** represent cgroup v2.

Create Tunnel Specifications

You can use one of the following tunnel types:

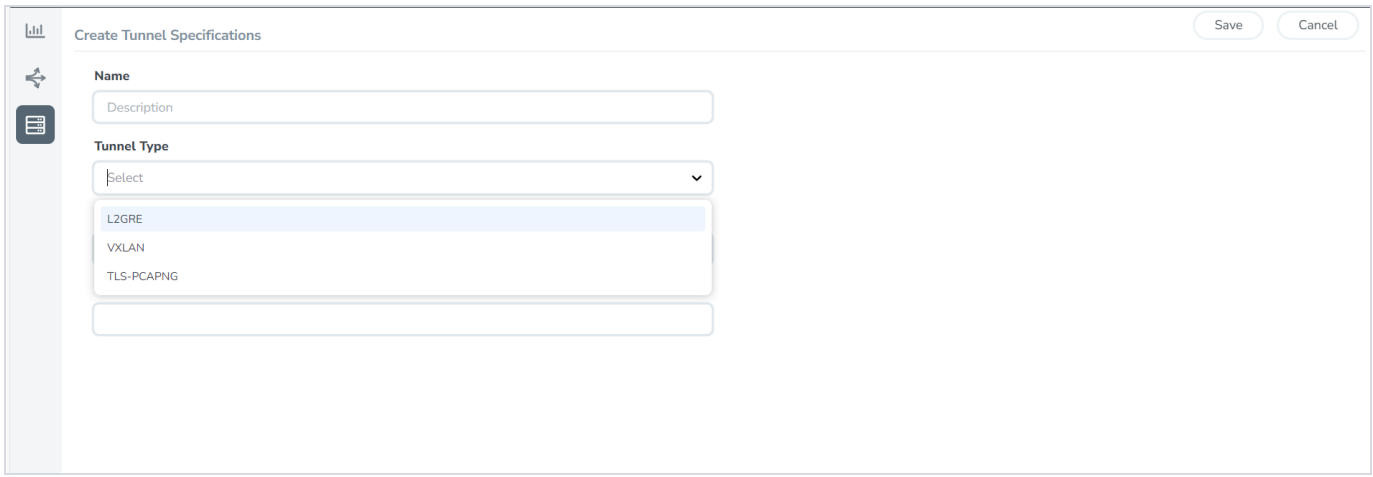
- L2GRE
- VXLAN
- TLS-PCAPNG.

The tunnel is an egress tunnel. For more information on creating a tunnel of type TLS-PCAPNG, refer to Secure Tunnels.

> **NOTE:** The L2GRE tunnel is not supported on the Azure platform.

To configure the tunnels:

1. Go to **Inventory** > **Resources > Tunnel Specifications**.
2. On the **Tunnel Specifications** page, navigate to the **Container** tab and select **Create**. The **Create Tunnel Specification** wizard appears.



3. Enter the name of the tunnel endpoint.

   > **NOTE:** Do not enter spaces in the alias name.

4. Select **L2GRE**, **VXLAN**, or **TLS-PCAPNG** tunnel type to create a tunnel.
5. Enter the IP address of the destination endpoint.
6. Enter a value for the tunnel key (applicable when the selected tunnel type is L2GRE).
7. Enter the identifier key for the VXLAN network (applicable when the selected tunnel type is VXLAN). The value ranges from 1 to (2^32) - 1.
8. Specify the destination port value.

9. Enter a value between 1 and 65535 (applicable when the selected tunnel type is VXLAN or TLS-PCAPNG).
10. Select **Save**.

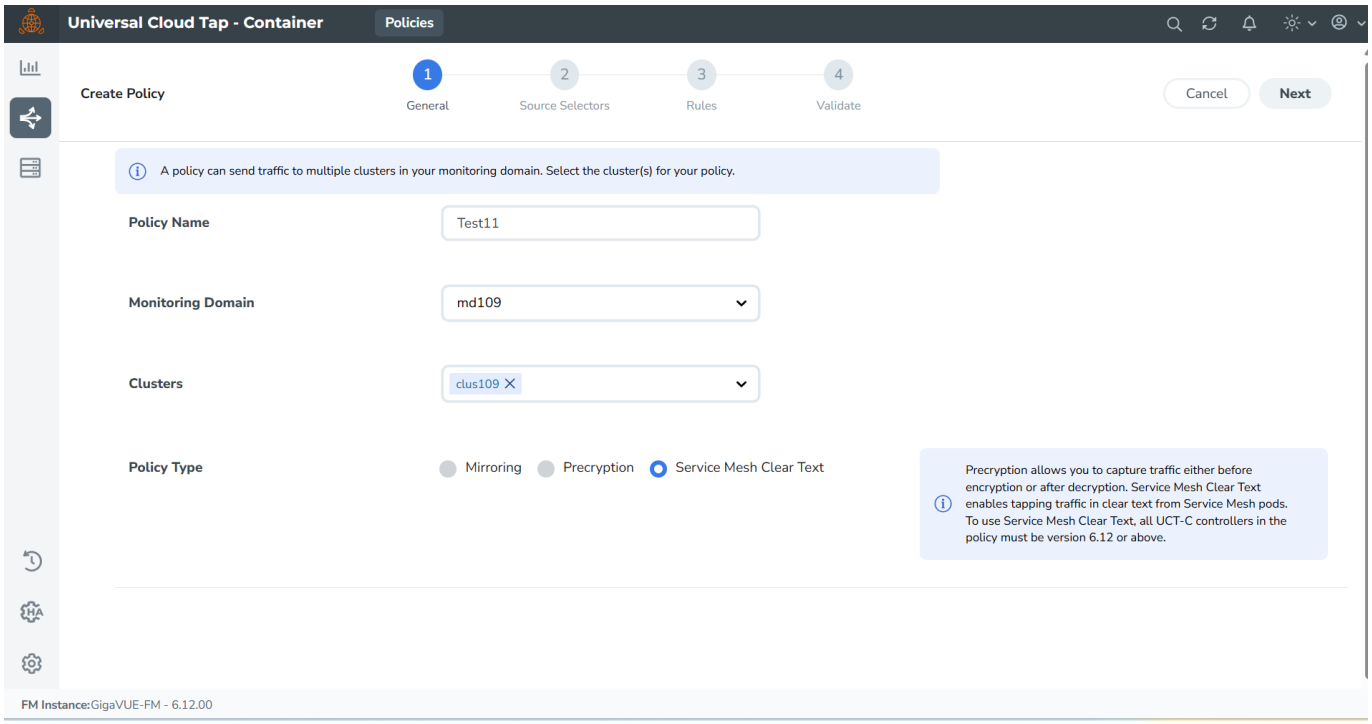
Configure Traffic Policy

The traffic from the workload pods is processed based on the Traffic Policy configuration. The UCT-C TAP routes the traffic to the tunnel destination IP addresses specified in the Traffic Policy rules.

You can refer to the GigaVUE API Reference for detailed information on the REST APIs of UCT-C.

To create UCT-C Traffic Policy in GigaVUE-FM, follow these steps:

1. Go to **Traffic > CONTAINER > Universal Cloud Tap - Container**.
   The **Policies** page appears.



2. In the **Policies** page, select **New**. The Create Policy wizard appears.

   NOTE: You can deploy a maximum of eight policies per Monitoring Domain.

3. In the **General** tab, enter a unique name for the Traffic Policy.
4. Select an existing Monitoring Domain.
   To create a new monitoring domain, refer to Create Monitoring Domain.
5. Select the required cluster from the drop-down menu.

6. Choose one of the following policy types based on your traffic visibility requirements:
    a. Mirroring – Captures and forwards all traffic as-is (default option).
    b. Precryption – Taps clear-text traffic before it is encrypted.
    c. Service Mesh Clear Text – Captures clear-text traffic from Service Mesh-enabled pods.
7. For details, refer to Configure Precryption in UCT-C.

> **Notes:**
>
> - Once the policy is deployed, you cannot change the Precryption Policy setting.
> - You can deploy only one Precryption policy per Monitoring Domain.
> - You can deploy only one Service Mesh Clear Text policy per Monitoring Domain.

8. Select **Next** to switch to the **Source Selectors** tab.
9. Select an existing source selector and select **Add** > **Create New**.
   To create a new source selector, refer to Create Source Selectors. You can configure a maximum of eight source selectors per policy.
10. In the **Source Selectors** page, select  to expand the **Default Exclusion** section.
    Default Exclusion Source Selector is applied automatically for all policies.
    - For Mirroring, you can exclude Service Mesh-enabled pods.
    - For Precryption, the system does not include Service Mesh-enabled pods.
    - For Service Mesh Clear Text, the system automatically includes only Service Mesh-enabled pods.

> **NOTE:** You can edit the values across the Monitoring Domain in **Inventory** > **Resources > Source Selectors** section. On the **Source Selectors Specifications** page, navigate to **Container > Default Exclusion**. In the **Edit Source Selector** wizard, you can edit the values in the **Exclusion Criteria** section.

11. Select **Next** to switch to the **Rules** tab.
    You must select CA in the Monitoring Domain page to use secure tunnels in rules.

NOTE: For Precryption and Service Mesh Clear Text, only one tunnel per policy is allowed. Rules follow the same structure as Precryption (passall, drop, or filtered rules).

12. Enter or select the required information for the **Ingress Rules** and the **Egress Rules** as described in the following steps:

    a. Select an existing tunnel or select **Create New**.
    For details, refer to Create Tunnel Specifications. For Precryption and Service Mesh clear text, only one Tunnel Specification field is displayed at the top for all the rules. For Mirroring, Tunnel Specification is configured for every individual rule.

    b. Enter a unique name for the rule.

    NOTE: We do not recommend Rule names ending with __I, __E, __RI, __RE as the names are invalid in policy rules. Rule names such as passall, ingress-passall, and egress-passall are restricted.

    c. Select the direction from the following options:

    - Bidirectional - Taps the traffic in both directions. Each bidirectional rule adds 2 ingress rules and 2 egress rules

    NOTE: When you apply filters to two pods on the same worker node to capture traffic in both directions, only one copy of the packet is tunneled out for each packet traveling from one pod to the other.

    - Ingress- Taps the ingress traffic
    - Egress - Taps the egress traffic
    - Ingress Pass All - Taps all the ingress traffic
    - Egress Pass All - Taps all the egress traffic

    NOTE: The maximum number of rules supported per direction is 32.

    d. Enter a priority value in the range 1-256 to specify the order of rule execution on the selected Pod. Unique Priority is enforced in a policy within ingress and egress space. Bidirectional rules get expanded in ingress and egress space. Priority is not applicable for Drop Rules. Drop Rules are executed first, followed by passall rules, and then filter rules based on specified priority values.

13. Select **Next** to switch to the **Validate** tab.

14. Select **Save** to the policy or click **Deploy**.
    The selected traffic policy rules are deployed to the required UCT-C TAPs present on the nodes corresponding to the source pods selected for monitoring.

NOTE: If policy deployment for pods fails with a **Duplicate Filter Rules** error, you need to decide which policy to apply to monitor the pod and remove the overlapping sources from the failed polcy.

View Policy Configurations

To view the Policy Configurations of the traffic policy configured in the GigaVUE-FM,

1. Go to **Traffic > CONTAINER > Universal Cloud Tap - Container**.

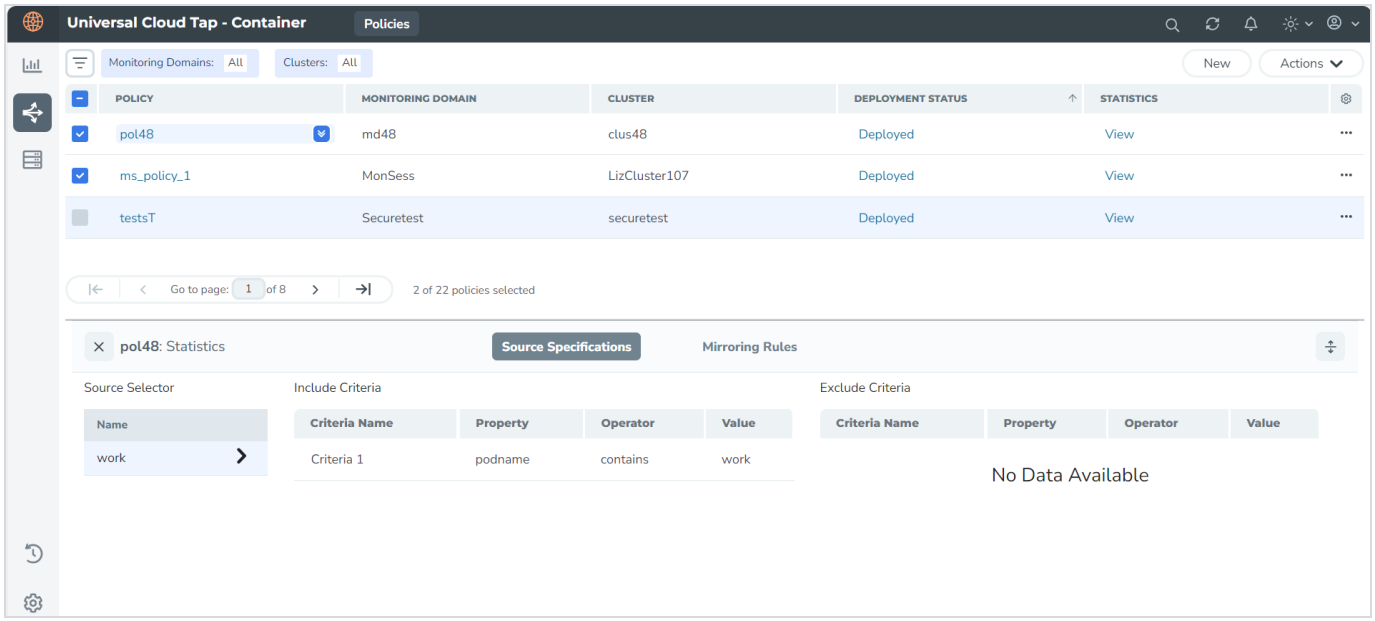
   The **Policies** page appears.

2. Select the required policy name.

   The configurations appear on the bottom of the **Policies** page. You can view the following tabs along with the policy name:

- Source Specifications
- Mirroring / Precryption / Service Mesh Rules

You can scroll each of the tables to view more columns. The fields and description for the tab are described in the respective topics.



Source Specifications

You can view the criteria based on which a pod is selected for tapping.

The fields and descriptions of the source specifications tab are described in the following table:

Table 1: Source Specifications

Tab- Source Specifications	Field	Description
Source Selector		
	Name	Specifies the name of the Source selector.
Include Criteria		
	Criteria Name	Specifies the include criteria for the source selector. Pod that matches the include criteria is part of the source for the given traffic policy.
	Property	Specifies the attributes of the pod. The available attributes are: • namespace • servicename • serviceip • podname • podip • podlables • nodename • nodepodcidr
	Operator	Specifies the operator used in the criteria.
	Value	Specifies the value for the attributes in the criteria.
Exclude Criteria		
	Criteria Name	Specifies the exclude criteria for the source selector. Pod that matches the exclude criteria will be excluded from the source for the given traffic policy.
	Property	Specifies the property in the exclude criteria based on which the pod associated with the source is excluded.
	Operator	Specifies the operator involved in the exclude criteria in tapping the traffic in the pod.
	Value	Specifies the value in the criteria based on which traffic in the pod is excluded.

Mirroring / Precryption / Service Mesh Rules

You can view the aggregate value of all the rules configured for the policy on the node in the UCT-C TAP present in a cluster. The fields and their descriptions in the Mirroring Rules tab are detailed in the following table:

Table 2: Mirroring Rules

Tab-Rules Rules	Field	Description
Mirroring / Precryption / Service Mesh Rules		
	Rule	Specifies the name of the rules in which the traffic is filtered in the pod. Click on the Rule name to view the filters.
	Tunnel	Specifies the tunnel details which is associated with the rules to send the traffic out. When you hover over the tunnel specification value, you can view the details of the tunnel in a message box.
	Priority	Specifies the priority assigned for the rule.
	Action	Specifies whether to pass or drop the rule.
	Direction	Specifies whether the traffic flow direction is ingress, egress, or bidirectional (both directions).
Filter		
	Type	Specifies the filter type.
	Filter	Specifies the name for the filter.
	Value	Specifies the value of the filter.

View Traffic Policy Statistics

Traffic Policy Statistics in the GigaVUE-FM provide visibility of the policies within a Monitoring Domain.

- When you deploy a policy on a worker Node, UCT-C TAP pods report policy statistics to GigaVUE-FM.
- These reports are generated at the pod level and include counter data for each policy.
- Policy statistics are sent to GigaVUE-FM every 5 minutes.
- GigaVUE-FM displays the statistics from the latest cycle received.

NOTE: Ensure that your UCT-C and GigaVUE-FM are time synchronized or configure NTP time synchronization.

The statistics counters reflect the activities of the deployed policies. They show how the policy statistics are directly related to the policy configured through GigaVUE-FM. For information regarding dashboard visualizations for Policy statistics, refer to Analytics Dashboard for UCT-C.

To view the policy configurations of the traffic policy configured in the GigaVUE-FM,

1. Go to **Traffic > CONTAINER > Universal Cloud Tap - Container**.
   The **Policies** page appears.
2. Select **View** status link in the **Statistics** column of a selected policy. The Policy Statistics appear at the bottom of the **Policies** page.

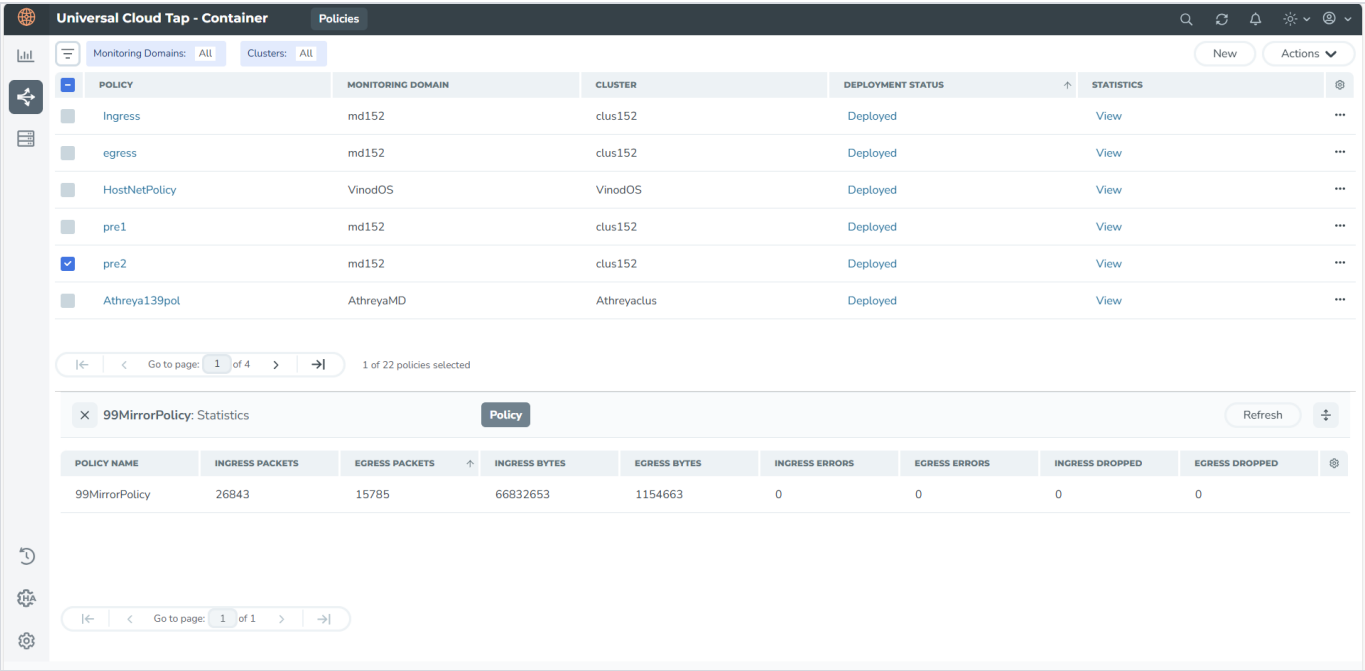


Table 3: Policy Statistics

Fields	Description
Policy Name	Name of the policy.
Ingress Packets	Total aggregate value of the ingress packets associated with the policy.
Egress Packets	Total aggregate value of the egress packets associated with the policy.
Ingress Bytes	Total aggregate value of the ingress bytes associated with the policy.
Egress Bytes	Total aggregate value of the egress bytes associated with the policy.
Ingress Errors	Total aggregate value of the ingress errors associated with the policy.
Egress Errors	Total aggregate value of the egress errors associated with the policy.
Ingress Dropped	Total aggregate value of the ingress packets dropped associated with the policy.
Egress Dropped	Total aggregate value of the egress packets dropped associated with the policy.

Configure UCT-C Features

This section walks you through enabling key UCT-C features to enhance traffic visibility and security.

For details, refer to the following sections:

- Configure Precryption in UCT-C
- Configure Secure Tunnels in UCT-C

Configure Precryption in UCT-C

GigaVUE-FM allows you to enable or disable the Precryption feature.

Rules and Notes

Memory limits for UCT-C:

- The memory limit changes depending on the number of vCPUs in the worker node. For example, if the worker node has 16 vCPUs, the Precryption feature consumes around 1GB of memory (16 * 64 MB).
- Secure tunnels require additional (16 *64 MB) memory. Hence, the total memory that you must allocate for the TAP is 1 GB.
- Use the PRECRYPTION_RING_BUFFER_MEMORY_MB field in the YAML file to configure the memory allocation.
- UCT-C supports both the protocol version IPv4 and IPv6.
- To use IPv6 tunnels, your GigaVUE-FM and the fabric components version be 6.6.00 or above.

The YAML configuration option allows you to choose the amount of buffer size.

To configure the Precryption feature in UCT-C, follow the steps listed in Configure Traffic Policy.

NOTE: You can deploy only one Precryption policy per Monitoring Domain.

After enabling the Precryption, configure the Create Source Selectors, and the **Rules**.

Selective Precryption

GigaVUE-FM allows you to filter packets during the Precryption in the Data Acquisition at the UCT-C level. This filtering is performed according to the L3/L4 5 tuple information (5-tuple filtering) running on the containers.

For details on how to configure Selective Precryption when configuring the **Rules**, refer to Enable Selective Precryption.

Enable Selective Precryption

If you wish to use selective Precryption, follow these steps:

1. Disable the **Enable** toggle button to turn off the default passall rule.
2. Select ⊕ to add another rule.

3. Enter the name of the rule and choose Pass (passes the traffic) or Drop (drops the traffic) in Action menu.

   > **NOTE:** In the absence of a Precryption rule, traffic is implicitly allowed. However, once rules are defined, they include an implicit pass all rule. If the traffic is not conformed to any of the specified rules, it will be passed.

4. Select one of the following directions:
   - **Bi-directional:** Allows the traffic in both directions of the flow. A single Bi-direction rule should consist of 1 Ingress and 1 Egress rule.
   - **Ingress:** Filters the traffic that flows in.
   - **Egress;** Filters the traffic that flows out.
5. Select the value of the priority.
   The value helps to prioritize the rules must for filtering.
   Select the value as 1 to pass or drop a rule in top priority.
   Similarly, you can select the value as 2, 3, 4 to 8, where 8 is for setting a rule with the least priority. You can add Drop rules based on the priority and, then add pass rules.
6. Select the required **Filter Type** (L3 or L4).
   a. For L3:
      i. Select the required **Filter Name**. The available options are IPv4 Source, IPv4 Destination, IPv6 Source, IPv6 Destination, and Protocol (common for both IPv4 and IPv6).
      ii. Enter or select the Filter Value based on the selected Filter Name.

      > **NOTE:** When using **Protocol** as the **Filter Name**, select **TCP** from the drop-down menu.

   b. For L4:
      i. Select the required **Filter Name**. The available options are Source Port and Destination Port.
      ii. Select the **Filter Relation**. The available options are Not Equal to and Equal to.
      iii. Enter the source or destination port value.

Configure Secure Tunnels in UCT-C

You can configure Secure tunnel on:

- Precryption Traffic
- Mirrored Traffic

Precryption Traffic

You can send the Precryption traffic through secure tunnel. When secure tunnel for Precryption is enabled, packets are framed and sent to the TLS socket. PCAPng format is used to send the packet.

When you enable the secure tunnel option for both regular and Precryption packets, two TLS secure tunnel sessions are created.

We recommend to enable secure tunnels for Precryption traffic to securely transfer the sensitive information.

For more information about PCAPng, refer to the section in GigaVUE V Series Applications Guide.

Mirrored Traffic

You can enable the Secure Tunnel for mirrored traffic. By default, Secure Tunnel is disabled.

Prerequisites

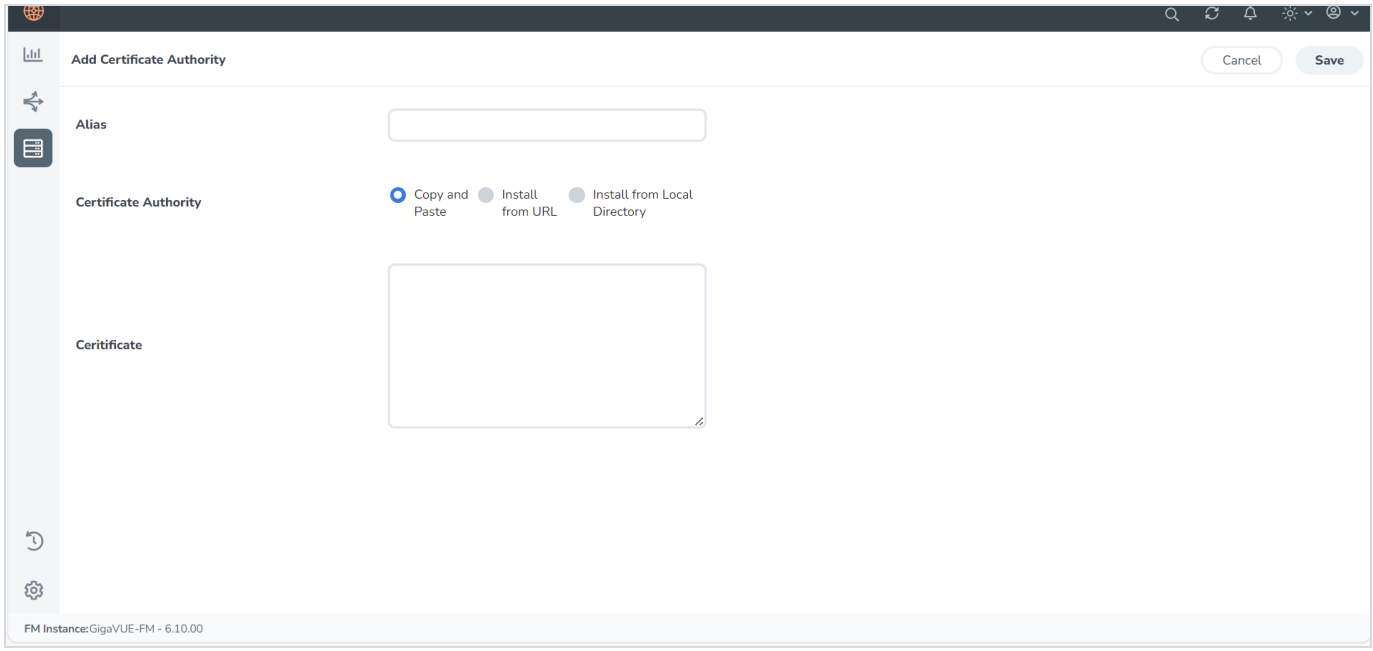
While creating Secure Tunnel, you must provide the following details:

- SSH key pair
- CA certificate

Configure Secure Tunnels from UCT-C Container to GigaVUE V Series Node

To configure a secure tunnel in a UCT-C Container, you must configure one end of the tunnel to the UCT-C and the other end to a GigaVUE Cloud Suite V Series node. You must configure CA certificates in UCT Container, and the private keys and SSL certificates in the GigaVUE Cloud Suite V Series Node. Refer to the following steps for configuration:

1. Upload a Custom Certificate:
    a. You must upload a CA to UCT-C Container for establishing a connection with the GigaVUE Cloud Suite V Series node.
    b. Go to **Inventory > Resources > Security > CA List**.
    c. Select **Add**.
    The **Add Certificate Authority** page appears.
    d. Enter the Alias and choose the certificate from the desired location.
    e. Select **Save**.
    For more information, refer to Adding Certificate Authority.



2. Upload an SSL Key: To add an SSL Key to GigaVUE Cloud Suite V Series node, follow the steps in the section *SSL Decrypt* in GigaVUE V Series Applications Guide.
3. Select the SSL Key when you create a Monitoring Domain and configure the fabric components in GigaVUE-FM.
4. Select the CA certificate when you create a monitoring domain and configuring the fabric components in GigaVUE-FM. To select the CA certificate, follow the steps in Create Monitoring Domain.
5. Create and add the secure tunnel when you configure the traffic policy. For details, refer to Configure Traffic Policy.

Adding Certificate Authority

The CA Certificate chain List page allows you to add the root CA for the devices.

To upload the CA Certificate chain using GigaVUE-FM, follow these steps:

1. Go to **Inventory > Resources > Security > CA List**.
2. Select **Add**, to add a new Custom Authority.
   The **Add Certificate Authority** page appears.
3. In the **Alias** field, enter the alias name of the CA Certificate chain Authority

4. Use one of the following options to enter the CA Certificate chain Authority:
   - **Copy and Paste**: In the **Certificate** field, enter the certificate.
   - **Install from URL**: In the **Path** field, enter the URL in the format: <protocol>://<username>@<hostname/IP address>/<file path>/<file name>. In the **Password** field, enter the password.
   - **Install from Local Directory**: Select **Choose File** to browse and select a certificate from the local directory.
5. Select **Save**.

Configure UCT-C Settings

You can configure the following UCT-C settings in GigaVUE-FM:

- UCT-C General Settings
- UCT-C Log Level Settings

UCT-C General Settings

In GigaVUE-FM, you can control the number of permitted clusters and purge time intervals of the UCT-C solution. You can specify the purge interval to automatically remove the UCT-Cs that are disconnected for a long duration.

To edit the UCT-C general settings:

1. In GigaVUE-FM, navigate to **Inventory > CONTAINER > Universal Cloud Tap - Container > Settings**.
   The **Settings** page appears with the existing General settings.
2. On the **General** section, select **Edit**.
3. Edit the following required values in the **General Settings** section.
   a. **Maximum number of clusters allowed**: Enter the maximum number of clusters allowed in the UCT-C solution. Enter a value between 1 and 64. The default value is 50.

   > **NOTE:** The maximum number of clusters allowed is based on the number of nodes and pods across the clusters. A maximum of 500 nodes or 25K pods are qualified across single or multiple clusters.

   b. **Disconnected UCT-C Purge Interval (days)**: Enter a value for the purge time interval for the disconnected UCT-Cs in days. Enter a value between 7 and 180. The default value is 30.
4. Select **Save**.
   The updates are visible on General Settings.

UCT-C Log Level Settings

In GigaVUE-FM, you can control the level of logs created at each individual UCT-C TAP and Controller for troubleshooting. The default UCT-C log file name format is as follows:

- For UCT-C TAP - **uctc_tap.log**
- For UCT-C Controller - **uctc_cntrl.log**

To view or edit the UCT-C log level settings:

1. For Log Level settings of **UCT-C TAP**:
   a. Go to **Inventory > CONTAINER > Universal Cloud Tap - Container > Nodes**, the **Nodes** page appears with the existing information about the node, cluster, pod, and UCT-C TAP details.
   b. On the **UCT-C TAP** section, on any Monitoring Domain, select the UCT-C TAP link that is in the connected status. The Universal Cloud Tap - Container: TAP Settings quick view appears.
   c. Edit the following required UCT-C log values in the **LOGGING** section.
      - Select the **Log Level** from the following options:
        - 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 severe error events that presumably lead the application to abort.
      - Enter a value for the number of lines in the **Log File Size** field.

        NOTE: On any of the above fields, click **Reset** to reset the value to default.
      - You can enable or disable statistics for the selected individual node using **Enable Policy Stats** option.
2. For Log Level settings of **UCT-C CONTROLLER**:
   a. Go to **Inventory > CONTAINER > Universal Cloud Tap - Container > Clusters**. The **Clusters** page appears with the existing information about the node, cluster, and UCT-C Controller details.

b. On the **UCT-C CONTROLLER** section, on any Monitoring Domain, select the UCT-C Controller link that is in the connected status.
The Universal Cloud Tap - Container: Controller Settings quick view appears.

c. Edit the following required UCT-C log values in the **LOGGING** section:

- Select the **Log Level** from the following options:
  - 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 severe error events that presumably lead the application to abort.
- Enter a value for the number of lines in the **Log File Size** field.

> **NOTE:** On any of the above fields, click **Reset** to reset the value to default.

Upgrade UCT-C

To upgrade UCT-C, you must perform the following steps:

1. **Upgrade to GigaVUE-FM 6.14**: Before upgrading GigaVUE-FM from earlier versions less than 6.8, delete the UCT-C Monitoring Domain, Policy, UCT-C Controller, and UCT-C TAP. Refer to GigaVUE-FM Installation and Upgrade guide for platform-specific upgrade instructions.

   > **NOTE:** GigaVUE-FM 6.14 is not compatible with older versions of UCT-C (less than 6.8). During the upgrade, GigaVUE-FM will delete all UCT-C Inventory and configurations.

2. **Upgrade to GigaVUE-FM (6.8+ to 6.14)**: When upgrading GigaVUE-FM from version 6.8 and above to 6.10, all the UCT-C configurations will be retained in GigaVUE-FM allowing for a direct upgrade.
3. **Upgrade to UCT-C 6.14**: Before upgrading UCT-C from earlier versions less than 6.8, you must delete the older versions and deploy UCT-C 6.14 version. To deploy UCT-C, refer to Steps to Delete and Redeploy the UCT-C Solution. Post UCT-C upgrade, verify that the controller version is 6.10.00.
4. **Upgrade to UCT-C (6.8+ to 6.14)**: When upgrading UCT-C from version 6.8 and above to 6.14, you should upgrade the UCT-C Controller first and then upgrade UCT-C TAP.

Notes:

- Changing the configuration from an IPv4 stack to an IPv6 stack or vice versa in the Helm chart is not supported through upgrades. If you need to switch between the IPv4 and IPv6 stacks, you must uninstall and then reinstall the Helm chart.
- During the upgrade from version less than 6.10, GigaVUE-FM will automatically remove the Host Network criteria from the DefaultExclusion Source Selector and will add the Host Network Enabled **Exclusion Criteria** by default to all the existing policies.

Post Upgrade Checklist

After upgrading, ensure that you verify the following interactions:

- Controller connectivity should be **Reachable**.
- Controller Heartbeat status should be **Connected**.
- TAP Heartbeat status should be **Connected**.
- Tools or V Series Node should receive traffic from monitored worker pods.
- Check for stale Kubernetes resources in the following components:
    1. **Cluster Role** - uctc-vpod-role
    2. **Secrets** - uctc-validator

  If the resources exist post upgrade, delete them using below:

```
kubectl delete clusterrole uctc-vpod-role
kubectl delete secrets  uctc-validator -n <namespace>
```

Steps to Delete and Redeploy the UCT-C Solution

Using Helm Charts

1. To uninstall the UCT-C Controller or TAP using Helm Charts, run the below command in the location where the UCT-C directory is present.

```
helm uninstall uctc -n <namespace>
```

2. To redeploy the UCT-C Controller or TAP using Helm Charts, edit the values (imagePullSecrets, namespace, GigaVUE-FM IP, external load balancer IP and Kubernetes API URL) in values.yaml file and run the below command in the location where the UCT-C directory is present.

```
helm install uctc <path_to_new_hemchart> -n <namespace>
```

3. You can alternatively run the following command to upgrade the UCT-C solution directly without deleting and redeploying.

```
helm upgrade uctc <path_to_new_hemchart> -n <namespace>
```

NOTE: If the Helm upgrade fails, you can try reverting to the previous version using Helm rollback. If the rollback also fails, delete the Helm and reinstall it.

For more details on the deployment of UCT-C Controller and TAPs using Helm Charts, refer to Helm Charts.

Analytics Dashboard for UCT-C

Analytics dashboards allow users to monitor the physical and virtual environment and detect anomalous behavior, and plan accordingly.

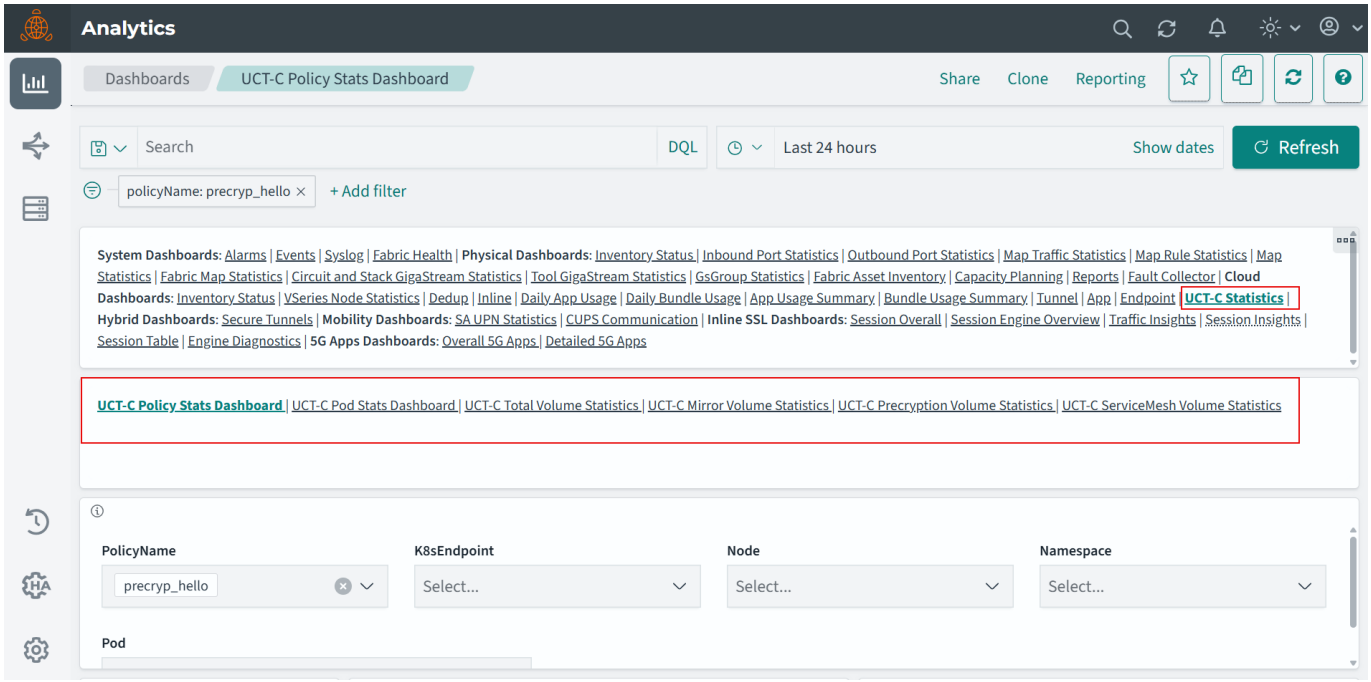
The UCT-C dashboard provides comprehensive insights into the Policy level, Pod level and Volume statistics. You can monitor both trend and total stats for pods associated with a selected policy and policies linked to a selected pod, within a specified time range.

How to Access the Dashboards

To access the UCT-C dashboards,

1. Go to  -> **Analytics -> Dashboards**.
2. Select **UCT-C Statistics** from the **System Dashboard** section.
3. The Dashboards page displays the following statistical dashboards.
   a. UCT-C Policy Statistics Dashboard
   b. UCT-C Pod Statistics Dashboard
   c. UCT-C Total Volume Statistics Dashboard
   d. UCT-C Mirror Volume Statistics Dashboard
   e. UCT-C Precryption Volume Statistics Dashboard
   f. UCT-C ServiceMesh Volume Statistics Dashboard

NOTE: Ensure that your UCT-C and GigaVUE-FM are time synchronized or configure NTP time synchronization.



UCT-C Policy Statistics Dashboard

1. On the **Dashboard**, select **UCT-C Policy Stats Dashboard** to view visualizations for Policy statistics.
2. From the drop-down menu, select the Policy name.
   The drop-down lists of **K8sEndpoint**, **Node**, **Namespace**, and **Pod**dynamically display the linked cluster, node, namespace, and pod details.
3. Select the required details to view the visualizations.

> **NOTE:** You can only select a maximum of 10 pods. You can maximize the visualizations panel and click **Add Filter** option to edit or add the Pod details.

UCT-C Policy Statistics- Displays the detailed visualizations for Policy level UCT-C statistics.

The following table lists the various Policy Statistics visualizations:

Table 4: UCT-C Policy Stats Dashboard

Visualizations	Details
UCT-C Policy Stats - Total Ingress Packets (Max 10)	Displays the total aggregate value of the ingress packets associated with the selected policy.
UCT-C Policy Stats - Total Egress Packets (Max 10)	Displays the total aggregate value of the egress packets associated with the selected policy.
UCT-C Policy Stats - Total Ingress Bytes (Max 10)	Displays the total aggregate value of the ingress bytes associated with the selected policy.

Visualizations	Details
UCT-C Policy Stats - Total Egress Bytes (Max 10)	Displays the total aggregate value of the egress bytes associated with the selected policy.
UCT-C Policy Stats - Total Ingress Dropped (Max 10)	Displays the total aggregate value of the ingress packets dropped associated with the selected policy.
UCT-C Policy Stats - Total Egress Dropped (Max 10)	Displays the total aggregate value of the egress packets dropped associated with the selected policy.
UCT-C Policy Stats - Total Ingress Errors (Max 10)	Displays the total aggregate value of the ingress errors associated with the selected policy.
UCT-C Policy Stats - Total Egress Errors (Max 10)	Displays the total aggregate value of the egress errors associated with the selected policy.
UCT-C Policy Stats - Ingress Packets Trend (Max 10)	Displays the Ingress Packets Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Egress Packets Trend (Max 10)	Displays the Egress Packets Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Ingress Bytes Trend (Max 10)	Displays the Ingress Bytes Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Egress Bytes Trend (Max 10)	Displays the Egress Bytes Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Ingress Dropped Trend (Max 10)	Displays the Ingress Dropped Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Egress Dropped Trend (Max 10)	Displays the Egress Dropped Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Ingress Errors Trend (Max 10)	Displays the Ingress Errors Trend details for each pod associated with the selected policy.
UCT-C Policy Stats - Egress Errors Trend (Max 10)	Displays the Egress Error Trend details for each pod associated with the selected policy.

UCT-C Pod Statistics Dashboard

1. On the **Dashboard**, select **UCT-C Pod Stats Dashboard** to view visualizations for Pod statistics
2. From the drop-down menu, select the Pod name.
   The drop-down lists of **K8sEndpoint** and **Node** filters that dynamically display the linked cluster, node, namespace, and pod details.
3. Select the required details to view the visualizations.

> **NOTE:** You can only select a maximum of 10 policies. You can maximize the visualizations panel and click **Add Filter** option to edit or add the Policy details.

UCT-C Pod Statistics- Displays the detailed visualizations for Pod level UCT-C statistics.

The following table lists the various Pod Statistics visualizations:

Table 5: UCT-C Pod Stats Dashboard

Visualizations	Details
UCT-C Pod Stats - Total Ingress Packets (Max 10)	Displays the total aggregate value of the ingress packets associated with the selected pod.
UCT-C Pod Stats - Total Egress Packets (Max 10)	Displays the total aggregate value of the egress packets associated with the selected pod.
UCT-C Pod Stats - Total Ingress Bytes (Max 10)	Displays the total aggregate value of the ingress bytes associated with the selected pod.
UCT-C Pod Stats - Total Egress Bytes (Max 10)	Displays the total aggregate value of the egress bytes associated with the selected pod.
UCT-C Pod Stats - Total Ingress Dropped (Max 10)	Displays the total aggregate value of the ingress packets dropped associated with the selected pod.
UCT-C Pod Stats - Total Egress Dropped (Max 10)	Displays the total aggregate value of the egress packets dropped associated with the selected pod.
UCT-C Pod Stats - Total Ingress Errors (Max 10)	Displays the total aggregate value of the ingress errors associated with the selected pod.
UCT-C Pod Stats - Total Egress Errors (Max 10)	Displays the total aggregate value of the egress errors associated with the selected pod.
UCT-C Pod Stats - Ingress Packets Trend (Max 10)	Displays the Ingress Packets Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Egress Packets Trend (Max 10)	Displays the Egress Packets Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Ingress Bytes Trend (Max 10)	Displays the Ingress Bytes Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Egress Bytes Trend (Max 10)	Displays the Egress Bytes Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Ingress Dropped Trend (Max 10)	Displays the Ingress Dropped Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Egress Dropped Trend (Max 10)	Displays the Egress Dropped Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Ingress Errors Trend (Max 10)	Displays the Ingress Errors Trend details for each policy associated with the selected pod.
UCT-C Pod Stats - Egress Errors Trend (Max 10)	Displays the Egress Error Trend details for each policy associated with the selected pod.

UCT-C Total Volume Statistics Dashboard

1. On the **Dashboard**, select **UCT-C Total Volume Statistics Dashboard** to view visualizations for Total Volume Statistics.
2. From the drop-down menu, select the K8sEndpoint that dynamically display the linked node details.
3. Select the required details to view the visualizations.

UCT-C Total Volume Statistics- Displays the detailed visualizations for Total VolumeUCT-C statistics.

The following table lists the various Total Volume Statistics visualizations:

Table 6: UCT-C Total Volume Statistics Dashboard

Visualizations	Details
UCT-C Total Volume Stats: Ingress Bits[Max 50]	Displays the total number of ingress bits for Mirror, Precryption, and Service Mesh traffic based on the chosen filter (Includes up to 50 nodes)
UCT-C Total Volume Stats: Egress Bits[Max 50]	Displays the total number of egress bits for Mirror, Precryption, and Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Total Volume Stats: Ingress Packets[Max 50]	Displays the total number of ingress packets for Mirror, Precryption, and Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Total Volume Stats: Egress Packets[Max 50]	Displays the total number of egress packets for Mirror, Pre-cryption, and Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror Volume Stats: Ingress Bits[Max 50]	Displays the total number of ingress bits for Mirror traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror Volume Stats: Egress Bits[Max 50]	Displays the total number of egress bits for Mirror traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror Volume Stats: Ingress Packets[Max 50]	Displays the total number of Ingress packets for Mirror traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror Volume Stats: Egress Packets[Max 50]	Displays the total number of egress packets for Mirror traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption Volume Stats: Ingress Bits[Max 50]	Displays the total number of ingress bits for Precryption traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption Volume Stats: Egress Bits[Max 50]	Displays the total number of egress bits for Precryption traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption Volume Stats: Ingress Packets[Max 50]	Displays the total number of Ingress packets for Precryption traffic based on the chosen filter (Includes up to 50 nodes).

Visualizations	Details
UCT-C Precryption Volume Stats: Egress Packets[Max 50]	Displays the total number of egress packets for Precryption traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh Volume Stats: Ingress Bits[Max 50]	Displays the total number of ingress bits for Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh Volume Stats: Egress Bits[Max 50]	Displays the total number of egress bits for Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh Volume Stats: Ingress Packets[Max 50]	Displays the total number of Ingress packets for Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh Volume Stats: Egress Packets[Max 50]	Displays the total number of egress packets for Service Mesh traffic based on the chosen filter (Includes up to 50 nodes).

UCT-C Mirror Volume Statistics Dashboard

1. On the **Dashboard**, select **UCT-C Mirror Volume Statistics Dashboard** to view visualizations for Mirror Volume Statistics.
2. From the drop-down menu, select the K8sEndpoint that dynamically display the linked node details.
3. Select the required details to view the visualizations.

UCT-C Mirror Volume Statistics - Displays the detailed visualizations for Mirror VolumeUCT-C statistics.

The following table lists the various Mirror Volume Statistics visualizations:

Table 7: UCT-C Mirror Volume Statistics Dashboard

Visualizations	Details
UCT-C Mirror: Total Rx Statistics [bps]	Displays the total ingress (Rx) bits per second(bps) mirror traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Rx Statistics per Worker node [bps]	Displays the total ingress (Rx) bits per second(bps) mirror traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Total Rx Statistics [pps]	Displays the total ingress (Rx) packets per second(pps) mirror traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Rx Statistics per Worker node [pps]	Displays the total ingress (Rx) packets per second(pps) mirror traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Total Tx Statistics [bps]	Displays the total egress (Tx) bits per second(bps) mirror traffic trend based on the chosen filter (Includes up to 50 nodes).

Visualizations	Details
UCT-C Mirror: Tx Statistics per Worker node [bps]	Displays the total egress (Tx) bits per second(bps) mirror traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Total Tx Statistics [pps]	Displays the total egress (Tx) packets per second(pps) mirror traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Mirror: Tx Statistics per Worker node [pps]	Displays the total egress (Tx) packets per second(pps) mirror traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).

UCT-C Precryption Volume Statistics Dashboard

1. On the **Dashboard**, select **UCT-C Precryption Volume Statistics Dashboard** to view visualizations for Precryption Volume Statistics.
2. From the drop-down menu, select the K8sEndpoint that dynamically display the linked node details.
3. Select the required details to view the visualizations.

UCT-C Precryption Volume Statistics Dashboard - Displays the detailed visualizations for Precryption VolumeUCT-C statistics.

The following table lists the various Precryption Volume Statistics visualizations:

Table 8: UCT-C Precryption Volume Statistics Dashboard

Visualizations	Details
UCT-C Precryption: Total Rx Statistics [bps]	Displays the total ingress (Rx) bits per second(bps) Precryption traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Rx Statistics per Worker node [bps]	Displays the total ingress (Rx) bits per second(bps) Precryption traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Total Rx Statistics [pps]	Displays the total ingress (Rx) packets per second(pps) Precryption traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Rx Statistics per Worker node [pps]	Displays the total ingress (Rx) packets per second(pps) Precryption traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Total Tx Statistics [bps]	Displays the total egress (Tx) bits per second(bps) Precryption traffic trend based on the chosen filter (Includes up to 50 nodes).

Visualizations	Details
UCT-C Precryption: Tx Statistics per Worker node [bps]	Displays the total egress (Tx) bits per second(bps) Precryption traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Total Tx Statistics [pps]	Displays the total egress (Tx) packets per second(pps) Precryption traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C Precryption: Tx Statistics per Worker node [pps]	Displays the total egress (Tx) packets per second(pps) precryption traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).

UCT-C ServiceMesh Volume Statistics Dashboard

1. On the **Dashboard**, select **UCT-C ServiceMesh Volume Statistics Dashboard** to view visualizations for Service Mesh volume statistics.
2. From the drop-down menu, select the K8sEndpoint that dynamically display the linked node details.
3. Select the required details to view the visualizations.

UCT-C ServiceMesh Volume Statistics Dashboard - Displays the detailed visualizations for ServiceMesh VolumeUCT-C statistics.

The following table lists the various ServiceMesh Volume Statistics visualizations:

Table 9: UCT-C ServiceMesh Volume Statistics Dashboard

Visualizations	Details
UCT-C ServiceMesh: Total Rx Statistics [bps]	Displays the total ingress (Rx) bits per second(bps) ServiceMesh traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Rx Statistics per Worker node [bps]	Displays the total ingress (Rx) bits per second(bps) ServiceMesh traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Total Rx Statistics [pps]	Displays the total ingress (Rx) packets per second(pps) ServiceMesh traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Rx Statistics per Worker node [pps]	Displays the total ingress (Rx) packets per second(pps) ServiceMesh traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Total Tx Statistics [bps]	Displays the total egress (Tx) bits per second(bps) ServiceMesh traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Tx Statistics per Worker node [bps]	Displays the total egress (Tx) bits per second(bps) ServiceMesh traffic trend per worker node based on the chosen filter (Includes up to 50

Visualizations	Details
	nodes).
UCT-C ServiceMesh: Total Tx Statistics [pps]	Displays the total egress (Tx) packets per second(pps) ServiceMesh traffic trend based on the chosen filter (Includes up to 50 nodes).
UCT-C ServiceMesh: Tx Statistics per Worker node [pps]	Displays the total egress (Tx) packets per second(pps) ServiceMesh traffic trend per worker node based on the chosen filter (Includes up to 50 nodes).

Troubleshoot UCT-C Issues

- **GigaVUE-FM to UCT-C Controller Connectivity Issues** - When UCT-C Controller connectivity is unreachable, verify whether **503 Service Temporarily Unavailable** error messages are observed in GigaVUE-FM's vmm.log (refer to the log messages below). If the error messages are available, check and update the UCT-C Controller service name or port number (as shown in the nginx.yaml) used in the ingress resource.

```
nginx.yaml
  apiVersion: networking.k8s.io/v1
  kind: Ingress
  metadata:
  annotations:
   kubernetes.io/ingress.allow-http: "false"
   kubernetes.io/ingress.class: nginx-uct
   nginx.ingress.kubernetes.io/backend-protocol: HTTPS
   nginx.ingress.kubernetes.io/configuration-snippet: proxy_set_header
  Authorization
   $http_authorization;
   nginx.ingress.kubernetes.io/rewrite-target: /
   nginx.ingress.kubernetes.io/secure-backends: "true"
   nginx.ingress.kubernetes.io/ssl-passthrough: "true"
  name: uct-cntlr-ingress
  namespace: uct
  spec:
   rules:
    - http:
       paths:
         - backend:
             service:
               name: gigamon-uctc-cntlr-service
                port:
                  number: 8443
  path: /
  pathType: ImplementationSpecific

  Log-Snippet
```

```
2024-08-16 08:00:35,527 INFO [uctcControllerConnectivity-585] UctcControllerRestClientImpl -
isAlive : connectivitUrl GET: https://<ExternalIP>:<ExternalPort>/api/v1.3/controller
2024-08-16 08:00:35,527 INFO [uctcControllerConnectivity-585] UctcRestClientBase - REQUEST GET
https://<ExternalIP>:<ExternalPort>/api/v1.3/controller null
```

```
2024-08-16 08:00:36,566 INFO [uctcControllerConnectivity-585] UctcRestTemplateResponseErrorHandler
- $$$ UCT-C 5XX Rest Error 503 Service Temporarily Unavailable
2024-08-16 08:00:36,566 ERROR [uctcControllerConnectivity-585] UctcRestClientBase - Request GET
https://<ExternalIP>:<ExternalPort>/api/v1.3/controller UctcRestException::
com.gigamon.cloud.uctc.rest.client.UctcRestException: UCTC SERVER
ERROR:: 503 SERVICE_UNAVAILABLE Service Temporarily Unavailable
```

- **UCT-C Controller not discovered by GigaVUE-FM** - When the UCT-C Controller is not discovered by GigaVUE-FM, check whether the Kubernetes cluster URL is updated properly in the yaml file.
- **UCT-C TAP not discovered by GigaVUE-FM**- When UCT-C tap is not discovered by GigaVUE-FM, verify whether the namespace in uctc-tap yaml file (as shown in the following uctc-tap.yaml) is same as that of UCT-C Controller yaml file.

  uct-tap.yaml

  ```
  # Value need to match me        tadata used for gcb-cntlr
  #  value: "<UCT-CNTLR-SVC-NAME.UCT-CNTLR-NAMESPACE>.svc.cluster.local"
  - name: UCTC_CNTLR_SVC_DNS
  value: gigamon-uctc-cntlr-service.<<namespace>>.svc.cluster.local  ===> This
  should be same as that of the namespace in which uctc-controller is deployed.
  ```

- **Policy Rules stuck in deploying status for nodes where UCT-C TAP pod is not present** - If Policy Source Selection Criteria matches Pods on the node where TAP is not launched, Rule status for those Pods will be 'deploying' until a UCT-C TAP pod gets launched on respective nodes. If Master Nodes in Cluster do not have UCT-C TAP, add nodename in the DefaultExclusion Source Selector.
  If you miss adding the node names, the policy rules on pods will be stuck in Undeploying status when you try to undeploy them. It is recommended that you delete the policy.
- To analyze and troubleshoot any additional issues, contact Contact Technical Support.

Policy Deployment Error Code

Error Code	Description
Pod doesn't exist.	Pod no longer exists in the cluster.
Number of network interfaces exceed max limit for a pod.	Number of network interfaces exceed max limit for a pod.
Policy not found for the pod.	Policy would have deleted from UCT-C Tap. Redeploy the policy.
Policy exceeds max limit for a pod.	Each Pod can have up to 8 polices.
Rule does not exist.	Rule does not exist in the UCT-C Tap. Redeploy the policy.
Duplicate Rule Exists for Policy.	When a new rule is pushed to TAP for a Pod which already has an identical rule (same Policy, same rule name, same filter and same tunnel). Remove duplicate filter and redeploy the policy.
Duplicate Rule Filter for pod.	Same filter is applied to a Pod from different rules

Error Code	Description
	(different policy name or rule name). Remove duplicate filter and redeploy the policy.
Rule exceeds max limit for a pod.	Each Pod interface can have up to 32 rules in each direction (ingress / egress).
Container doesn't exist.	TAP failed retrieving container info by container id.
Policy deployment not allowed for Hostnet pods with cgroupv1 .	Policy deployment not allowed for Hostnet pods with cgroupv1. cgroup v2 is required on the worker node to support the Host Network Enabled feature.

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 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 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 Cloud Suite devices; reference information and specifications for the respective GigaVUE Cloud Suite 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 Cloud Suite 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 Cloud Suite 6.14 Hardware and Software Guides
GigaVUE Cloud Suite Deployment Guide Universal Cloud Tap - Container
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)