



# Universal Cloud Tap- Container Deployment Guide

**GigaVUE Cloud Suite**

Product Version: 6.9

Document Version: 1.0

(See Change Notes for document updates.)

**Copyright 2024 Gigamon Inc. All rights reserved.**

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

**Trademark Attributions**

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

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

# Change Notes

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

Product Version	Document Version	Date Updated	Change Notes
6.9	1.0	12/06/2024	The original release of this document with 6.9.00 GA.

# Contents

<b>Universal Cloud Tap-Container Deployment Guide</b> .....	<b>1</b>
Change Notes .....	3
Contents .....	4
<b>Universal Cloud Tap - Container</b> .....	<b>6</b>
<b>Components of Universal Cloud Tap - Container</b> .....	<b>6</b>
<b>Architecture of Universal Cloud Tap - Container</b> .....	<b>7</b>
<b>UCT-C Solution and GigaVUE-FM Interaction</b> .....	<b>8</b>
UCT-C Controller and TAP Registration .....	9
UCT-C Controller and TAP Deregistration .....	9
UCT-C Controller and TAP Heartbeats .....	9
UCT-C Controller Connectivity .....	10
Monitoring Domain and Traffic Policy .....	10
<b>Get Started with Universal Cloud Tap - Container</b> .....	<b>10</b>
Pre-requisites of UCT-C .....	11
License Information .....	11
Network Firewall Requirements .....	12
Compute Requirements .....	17
Kernel and CPU Requirements for Universal Cloud Tap - Container .....	18
Supported Platforms for UCT-C .....	18
Security Requirements .....	19
<b>Configure Universal Cloud Tap - Container</b> .....	<b>20</b>
Create Secret to pull UCT-C image .....	20
Deploy UCT-C Controller and TAP in Kubernetes .....	21
Deploy UCT-C Controller and TAPs .....	21
Deploy UCT-C on Redhat Openshift Platform using Openshift UI .....	28
Configure UCT-C Controller and TAP through GigaVUE-FM .....	30
Source Specifications .....	42
Mirroring Rules or Precryption Rules .....	43
Precryption™ .....	45
Secure Tunnels .....	52
Adding Certificate Authority .....	55
<b>Configure UCT-C Settings</b> .....	<b>56</b>
UCT-C General Settings .....	56

- UCT-C Log Level Settings .....57
- Upgrade UCT-C ..... 57**
- Post Upgrade Checklist .....58
- Steps to Delete and Redeploy the UCT-C Solution .....59
  - Using YAML files ..... 59
  - Using Helm Charts ..... 59
- Debuggability and Troubleshooting ..... 60**
- Additional Sources of Information ..... 62**
- Documentation ..... 62
  - How to Download Software and Release Notes from My Gigamon ..... 64
- Documentation Feedback ..... 65
- Contact Technical Support ..... 66
- Contact Sales ..... 66
  - Premium Support .....67
- The VÜE Community .....67
- Glossary ..... 68**

# Universal Cloud Tap - Container

Universal Cloud Tap - Container (UCT-C) earlier known as Universal Container Tap (UCT) is a containerized component that provides the network broker features in a containerized form. UCT-C can perform traffic acquisition, basic filtering, and tunneling support. UCT-C is deployed as a Pod in the given worker node where the workloads are running.

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

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 the 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.

This guide provides an overview of Universal Cloud Tap - Container and describes how to install and deploy UCT-C components in your Pods.

Topics:

- [Architecture of Universal Cloud Tap - Container](#)
- [UCT-C Solution and GigaVUE-FM Interaction](#)
- [Get Started with Universal Cloud Tap - Container](#)
- [Configure Universal Cloud Tap - Container](#)
- [Configure UCT-C Settings](#)
- [Precryption™](#)
- [Secure Tunnels](#)

## Components of Universal Cloud Tap - Container

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

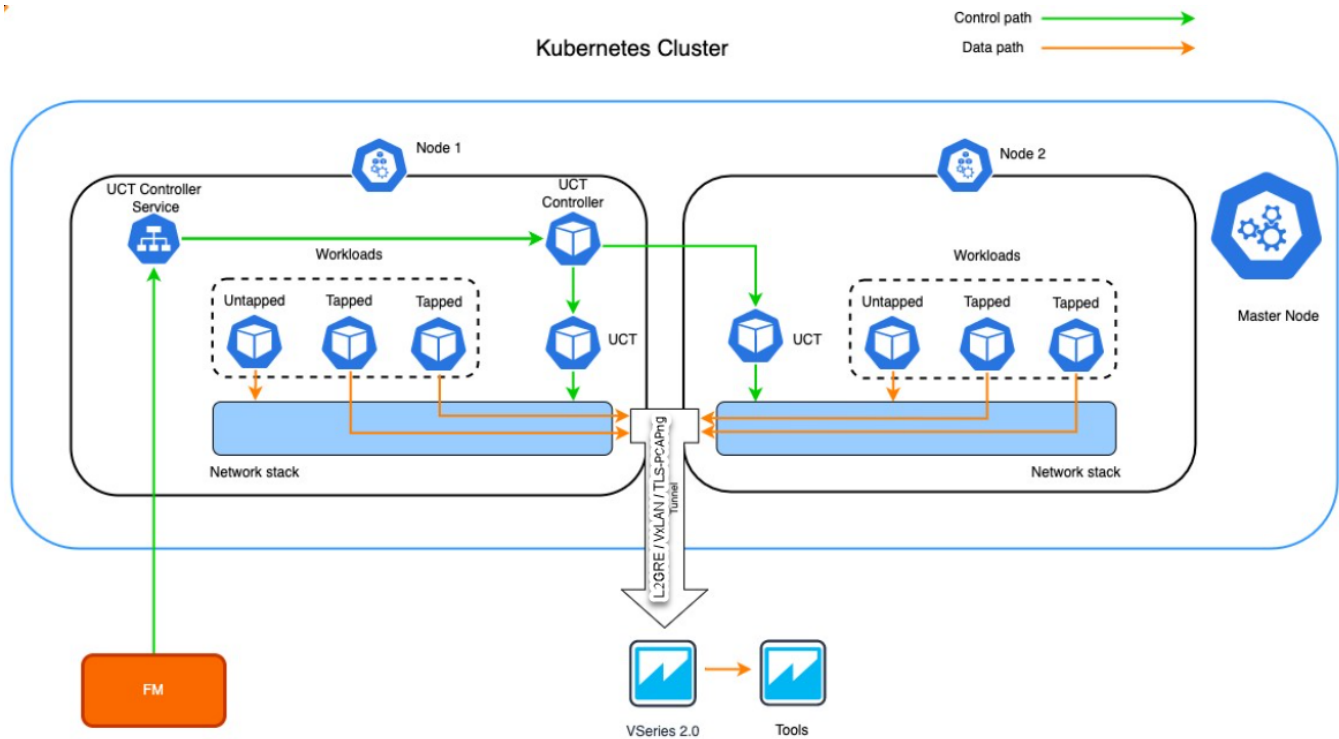
- **GigaVUE-FM fabric manager** is a web-based fabric management and orchestration interface that provides a single pane of glass visibility, management, and orchestration of both the physical and virtual traffic that form the UCT-C.
- **UCT-C Tap** is the primary UCT-C module that collects the workload traffic, filters the traffic, and tunnels the filtered traffic directly to the tools or through the GigaVUE V Series Nodes. UCT-C Tap also sends the traffic policy statistics and heartbeats to 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.

## Architecture of Universal Cloud Tap - Container

The following diagram illustrates the architecture of Universal Container Tap environment.



1. UCT-C Controller registers with GigaVUE-FM.
2. The UCT-C Tap is registered with GigaVUE-FM through the UCT-C Controller
3. GigaVUE-FM deploys the traffic policy on the UCT-Cs.  
Communication of configuration, data, and statistics to and from UCT-C is performed through the UCT-C Controller Service. GigaVUE-FM communicates with the UCT-C Taps through the UCT-C Controller. GigaVUE-FM deploys the traffic policy on UCT-C and receives the statistics from UCT-C tap through UCT-C controller.
4. The filtered network packets are tunneled directly to the Tools or through the GigaVUE V Series nodes running on any supported GigaVUE Cloud Suite in the cloud environment.
5. The UCT-C Controller collects the data from UCT-C Taps and sends the collected statistics and heartbeats to GigaVUE-FM.

## UCT-C Solution and GigaVUE-FM Interaction

The following are the interactions between the UCT-C solution and GigaVUE-FM:



- [UCT-C Controller and TAP Registration](#)
- [UCT-C Controller and TAP Deregistration](#)
- [UCT-C Controller and TAP Heartbeats](#)
- [UCT-C Controller Connectivity](#)
- [Monitoring Domain and Traffic Policy](#)

## UCT-C Controller and TAP Registration

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 Firewall Requirements](#).

UCT-C supports IPv4 and IPv6 protocols. For more information, refer to [Deploy UCT-C Controller and TAP in Kubernetes](#).

## UCT-C Controller and TAP Deregistration

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.

## UCT-C Controller and TAP Heartbeats

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.

## 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.

## Monitoring Domain and Traffic Policy

You can configure and manage the Monitoring Domains, Clusters, Source Inventories, and Traffic Policies of UCT-C in GigaVUE-FM. For more information, refer to [Configure UCT-C Controller and TAP through GigaVUE-FM](#).



- A Traffic Policy is a combination of Source Selection, Rules and Tunnels.
- The Source Selector determines the target pods.
- Rules filter the traffic from the source pods selected by the Source Selector criteria.
- Tunnel is the destination where the traffic matching rule will be routed

# Get Started with Universal Cloud Tap - Container

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

Refer to the following sections for details:

- [Pre-requisites of UCT-C](#)
- [Components of Universal Cloud Tap - Container](#)
- [License Information](#)
- [Network Firewall Requirements](#)
- [Compute Requirements](#)

## Pre-requisites of UCT-C

The following are the pre-requisites of UCT-C:

- GigaVUE-FM and Kubernetes Cluster should be up and running.
- GigaVUE-FM and Kubernetes Cluster should be able to communicate with each other.
- To work with UCT-C, you must have knowledge of the following platforms and products:
  - a. Kubernetes
  - b. GigaVUE-FM
  - c. GigaVUE V Series
  - d. Knowledge of working in any platform, such as AWS EKS, Azure AKS, and RedHat OpenShift where the Kubernetes clusters are deployed.
- To deploy UCT-C using YAML files and Helm charts, ensure that you install the following:
  - Kubectl to deploy using YAML files
  - Helm to deploy using Helm charts (single and dual)

For more information on deployment of UCT-C Controller and Taps using the YAML files or the Helm Charts, refer to the following sections in [Deploy UCT-C Controller and TAPs](#) topic:

- [Using YAML files](#)
- [Using Helm Charts](#)

**NOTE:** Traffic can be terminated directly to the tool or can be directed to GigaVUE V Series node, where it can be filtered before sending to the tool.

Refer to the following topics for more detailed information:

- [License Information](#)
- [Network Firewall Requirements](#)
- [Compute Requirements](#)
- [Kernel and CPU Requirements for Universal Cloud Tap - Container](#)
- [Supported Platforms for UCT-C](#)

## 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 licensing has a service period of 1 month. Service period is the period of time for which the total usage or overage is tracked.

To purchase licenses with the Volume-Based License (VBL) option, contact our Sales. Refer to [Contact Sales](#) for more details.

## Network Firewall Requirements

The following table lists the Network Firewall / Security Group requirements for GigaVUE Cloud Suite.

**NOTE:** When using dual stack network, the below mentioned ports must be opened for both IPv4 and IPv6.

GigaVUE-FM				
Direction	Protocol	Port	Source CIDR	Purpose
Inbound	TCP	443	Administrator Subnet	Allows GigaVUE-FM to accept Management connection using REST API. Allows users to access GigaVUE-FM UI securely through an HTTPS connection.
Inbound	TCP	22	Administrator Subnet	Allows CLI access to user-initiated management and diagnostics.
Inbound (This is the port used for Third Party Orchestration)	TCP	443	UCT-V Controller IP	Allows GigaVUE-FM to receive registration requests from UCT-V Controller using REST API.
Inbound (This is the port used for Third Party Orchestration)	TCP	443	GigaVUE V Series Node IP	Allows GigaVUE-FM to receive registration requests from GigaVUE V Series Node using REST API when GigaVUE V Series Proxy is not used.
Inbound (This is the port used for Third Party Orchestration)	TCP	443	GigaVUE V Series Proxy IP	Allows GigaVUE-FM to receive registration requests from GigaVUE V Series Proxy using REST API.

Inbound	TCP	443	UCT-C Controller IP	Allows GigaVUE-FM to receive registration requests from UCT-C Controller using REST API.
Inbound	TCP	5671	GigaVUE V Series Node IP	Allows GigaVUE-FM to receive traffic health updates from GigaVUE V Series Nodes.
Inbound	TCP	5671	UCT-V Controller IP	Allows GigaVUE-FM to receive statistics from UCT-V Controllers.
Inbound	TCP	5671	UCT-C Controller IP	Allows GigaVUE-FM to receive statistics from UCT-C Controllers.
Inbound	UDP	2056	GigaVUE V Series Node IP	Allows GigaVUE-FM to receive Application Intelligence and Application Visualization reports from GigaVUE V Series Node.
Direction	Protocol	Port	Destination CIDR	Purpose
Outbound	TCP	9900	GigaVUE-FM IP	Allows GigaVUE-FM to communicate control and management plane traffic with UCT-V Controller.
Outbound (optional)	TCP	8890	GigaVUE V Series Proxy IP	Allows GigaVUE-FM to communicate control and management plane traffic to GigaVUE V Series Proxy.
Outbound	TCP	8889	GigaVUE V Series Node IP	Allows GigaVUE-FM to communicate control and management plane traffic to GigaVUE V Series Node.
Outbound	TCP	8443 (default)	UCT-C Controller IP	Allows GigaVUE-FM to communicate control and management plane traffic to UCT-C Controller.
Outbound	TCP	443	Any IP Address	Allows GigaVUE-FM to reach the Public Cloud Platform APIs.
UCT-V Controller				
Direction	Protocol	Port	Source CIDR	Purpose
Inbound	TCP	9900	GigaVUE-FM IP	Allows UCT-V Controller to communicate control and management plane traffic with GigaVUE-FM
Inbound	TCP	9900	UCT-V or Subnet IP	Allows UCT-V Controller to receive traffic health updates from UCT-V.

Inbound (This is the port used for Third Party Orchestration)	TCP	8891	UCT-V or Subnet IP	Allows UCT-V Controller to receive the registration requests from UCT-V.
Inbound	TCP	22	Administrator Subnet	Allows CLI access for user-initiated management and diagnostics, specifically when using third party orchestration.
Direction	Protocol	Port	Destination CIDR	Purpose
Outbound (This is the port used for Third Party Orchestration)	TCP	443	GigaVUE-FM IP	Allows UCT-V Controller to send the registration requests to GigaVUE-FM using REST API.
Outbound	TCP	9901	UCT-V Controller IP	Allows UCT-V Controller to communicate control and management plane traffic with UCT-Vs.
Outbound	TCP	5671	GigaVUE-FM IP	Allows UCT-V Controller to send traffic health updates to GigaVUE-FM.
UCT-V				
Direction	Protocol	Port	Source CIDR	Purpose
Inbound	TCP	9901	UCT-V Controller IP	Allows UCT-V to receive control and management plane traffic from UCT-V Controller
Direction	Protocol	Port	Destination CIDR	Purpose
Outbound (This is the port used for Third Party Orchestration)	TCP	8891	UCT-V Controller IP	Allows UCT-V to communicate with UCT-V Controller for registration and Heartbeat
Outbound	UDP (VXLAN)	VXLAN (default 4789)	GigaVUE V Series Node IP	Allows UCT-V to tunnel VXLAN traffic to GigaVUE V Series Nodes
Outbound	IP Protocol (L2GRE)	L2GRE (IP 47)	GigaVUE V Series Node IP	Allows UCT-V to tunnel L2GRE traffic to GigaVUE V Series Nodes
Outbound (Optional - This port is used only for Secure Tunnels)	TCP	11443	GigaVUE V Series Node IP	Allows UCT-V to securely transfer the traffic to the GigaVUE V Series Node
Outbound	TCP	9900	UCT-V Controller IP	Allows UCT-V to send traffic health updates to UCT-V

				Controller.
<b>GigaVUE V Series Node</b>				
Direction	Protocol	Port	Source CIDR	Purpose
Inbound	TCP	8889	GigaVUE-FM IP	Allows GigaVUE V Series Node to communicate control and management plane traffic with GigaVUE-FM
Inbound	TCP	8889	GigaVUE V Series Proxy IP	Allows GigaVUE V Series Node to communicate control and management plane traffic with GigaVUE V Series Proxy.
Inbound	UDP (VXLAN)	VXLAN (default 4789)	UCT-V Subnet IP	Allows GigaVUE V Series Nodes to receive VXLAN tunnel traffic to UCT-V
Inbound	IP Protocol (L2GRE)	L2GRE	UCT-V Subnet IP	Allows GigaVUE V Series Nodes to receive L2GRE tunnel traffic to UCT-V
Inbound	UDPGRE	4754	Ingress Tunnel	Allows GigaVUE V Series Node to receive tunnel traffic from UDPGRE Tunnel
Inbound	TCP	22	Administrator Subnet	Allows CLI access for user-initiated management and diagnostics, specifically when using third party orchestration.
Inbound (Optional - This port is used only for Secure Tunnels)	TCP	11443	UCT-V subnet	Allows to securely transfer the traffic to GigaVUE V Series Nodes.
Inbound (Optional - This port is used only for configuring AWS Gateway Load Balancer)	UDP (GENEVE)	6081	Ingress Tunnel	Allows GigaVUE V Series Node to receive tunnel traffic from AWS Gateway Load Balancer.
Direction	Protocol	Port	Destination CIDR	Purpose
Outbound	TCP	5671	GigaVUE-FM IP	Allows GigaVUE V Series Node to send traffic health updates to GigaVUE-FM.
Outbound	UDP (VXLAN)	VXLAN (default 4789)	Tool IP	Allows GigaVUE V Series Node to tunnel output to the tool.
Outbound	IP Protocol (L2GRE)	L2GRE (IP 47)	Tool IP	Allows GigaVUE V Series Node to tunnel output to the tool.
Outbound	UDP	2056	GigaVUE-FM IP	Allows GigaVUE V Series Node

				to send Application Intelligence and Application Visualization reports to GigaVUE-FM.
Outbound	UDP	2055	Tool IP	Allows GigaVUE V Series Node to send NetFlow Generation traffic to an external tool.
Outbound	UDP	514	Tool IP	Allows GigaVUE V Series Node to send Application Metadata Intelligence log messages to external tools.
Bidirectional (optional)	ICMP	<ul style="list-style-type: none"> <li>echo request</li> <li>echo reply</li> </ul>	Tool IP	Allows GigaVUE V Series Node to send health check tunnel destination traffic.
Outbound (This is the port used for Third Party Orchestration)	TCP	8891	GigaVUE V Series Proxy IP	Allows GigaVUE V Series Node to send registration requests and heartbeat messages to GigaVUE V Series Proxy when GigaVUE V Series Proxy is used.
Outbound (This is the port used for Third Party Orchestration)	TCP	443	GigaVUE-FM IP	Allows GigaVUE V Series Node to send registration requests and heartbeat messages to GigaVUE-FM when GigaVUE V Series Proxy is not used.
Outbound (Optional - This port is used only for Secure Tunnels)	TCP	11443	Tool IP	Allows to securely transfer the traffic to an external tool.
<b>GigaVUE V Series Proxy (optional)</b>				
Direction	Protocol	Port	Source CIDR	Purpose
Inbound	TCP	8890	GigaVUE-FM IP	Allows GigaVUE-FM to communicate control and management plane traffic with GigaVUE V Series Proxy.
Inbound (This is the port used for Third Party Orchestration)	TCP	8891	GigaVUE V Series Node IP	Allows GigaVUE V Series Proxy to receive registration requests and heartbeat messages from GigaVUE V Series Node.
Inbound	TCP	22	Administrator Subnet	Allows CLI access for user-initiated management and diagnostics, specifically when using third party orchestration.
Direction	Protocol	Port	Destination CIDR	Purpose
Outbound	TCP	443	GigaVUE-FM IP	Allows GigaVUE V Series Proxy to communicate the



				registration requests to GigaVUE-FM
Outbound	TCP	8889	GigaVUE V Series Node IP	Allows GigaVUE V Series Proxy to communicate control and management plane traffic with GigaVUE V Series Node
<b>Universal Cloud Tap - Container</b> 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.
<b>UCT-C Controller</b> 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.
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 should be opened on GigaVUE-FM:

Direction	Port	Purpose
Inbound	443	GigaVUE-FM REST service port.
Outbound	4433	Allows GigaVUE-FM to communicate with UCT-C Controller.
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 table 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 kernel requirements for different platforms are as follows:

- **AWS/EKS/Azure AKS/Native Kubernetes** - 5.4
- **GKE** - 5.15 (Ubuntu OS)
- **VMware Tanzu Photon-** 4.19 (Photon+ OS)

The minimum 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

## Supported Platforms for UCT-C

The following tables list the different platforms and their Kubernetes version, Container Runtime Interface (CRI), and Container Network Interface (CNI).

**NOTE:** As an end user, you must have an understanding and knowledge of your container services.

Platform	Kubernetes Version	CRI	CNI
Amazon Elastic Kubernetes Service (EKS)	1.26, 1.27, 1.28	Containerd	VPC
Rancher	1.23 to 1.29	Containerd, CRI-dockerd	Calico, Flannel, Canal, Weave
Red Hat OpenShift	4.13 and 4.14	CRI-O	SDN, OVN
Kubernetes	1.26 to 1.30	Containerd	Flannel, Cilium, Calico, Multus

**NOTE:** UCT-C is platform and CNI independent software. It is intended to function correctly on other Kubernetes platforms such as Azure Kubernetes Service (AKS), GKE, and VMware Tanzu. If you have any issues or need further assistance, please contact [Gigamon Technical Support](#).

## Security Requirements

To create a Service Account with privileged access for tapping, run the below listed commands

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

Use the Gigamon service account in uctc-tap.yaml, which allows uctc-tap to come up as a privileged pod.

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

1. **Using YAML:** In YAML deployment, for the Red Hat OpenShift Container Platform, use the below command.

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

2. **Using Helm:** In a Helm deployment for the Red Hat OpenShift Container Platform, set the create value to "**True**" under security Context Constraints in values.yaml. This will generate a customized security context with the permissions required for deploying the UCT-C Solution in OpenShift.

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

# Configure Universal Cloud Tap - Container

Setting up UCT-C involves the following two steps:

- [Deploy UCT-C Controller and TAP in Kubernetes](#)
- [Configure UCT-C Controller and TAP through GigaVUE-FM](#)



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

## Create Secret to pull UCT-C image

To create secret, follow these steps:

1. UCT-C Controller and Tap images will be available for each and every release build in the Gigamon software portal. Download the respective UCT-C release build from the repository and untar the file. After you finish untarring the file, you can extract the images and import them to your repository.

**NOTE:** [Contact Technical Support](#) or [Contact Sales](#) for information on downloading the respective UCT-C build from the Gigamon software portal.

2. Get the **username/key/password** from support team and encode using the command:

```
echo -n <<user name>>:<<keys>> | base64
```

3. Update the keys in the following content and save it as a JSON file:

```
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "<<keys>>"
    }
  }
}
```

4. Create secret using the following command (regcred is used as name of the secret in this doc):

```
kubectl create secret generic <<name of the secret>> --from-file=.dockerconfigjson=<<json file path >>
```

```
--type=kubernetes.io/dockerconfigjson -n <<namespace where UCT-C Controller and tap will be deployed>>
```

## Deploy UCT-C Controller and TAP in Kubernetes

To fully deploy UCT-C, the following steps are required to be completed:

1. Implement external access to the Kubernetes environment (e.g., ingress, external public IPs, load balancers) to allow communication between the UCT-C Controller and GigaVUE-FM.
2. Ensure that the firewall rules on Kubernetes nodes are met according to the [Network Firewall Requirements](#).
3. YAML files or HELM charts are available as part of the UCT-C images. You should untar the UCT-C image to access the readme files.
4. Add the UCT-C images to a private Docker registry or ensure that the files can be pulled from the Docker Hub registry. You can spin up or spin down the UCT-C instances based on your traffic load.
5. Deploy UCT-C Controller and Taps using YAML files or Helm Charts. Refer to [Deploy UCT-C Controller and TAPs](#).

### Deploy UCT-C Controller and TAPs

You can deploy the UCT-C Controller and TAPs using the YAML files or the Helm Charts. Refer to the following sections for detailed information.

- [Using YAML files](#)
- [Using Helm Charts](#)

#### Using YAML files

YAML files will be available for each and every release build in the Gigamon software portal. Download the respective UCT-C release build from the repository and untar the **.tgz** file. After you finish untarring the file, you can extract the YAML file and further update the following fields in **uctc-controller.yaml** and **uctc-tap.yaml** file.

**NOTE:** [Contact Technical Support](#) or [Contact Sales](#) for information on downloading the respective UCT-C build from the Gigamon software portal.

#### Deployment of UCT-C Controller

Follow these steps to deploy the UCT-C Controller:

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

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

After extracting the tar file, navigate to the YAML folder in the newly created **uctc-cntlr-<image version>-<build number>** folder and update the details given in the below steps.

- Provide the created **secret** in the following section of the YAML file:

```
imagePullSecrets:
  - name: <secret>
```

- Provide the **FM IP address** in the following section of the YAML file:

```
command:
  - /uct-controller
  - <FM IPv4 or FM IPv6 or FM IPv4,FM IPv6(both with comma-separation notation)>
  - '443'
  - '8443'
  - '0'
  - "/etc/gcbcerts"
  - "gcb-cert.pem"
  - "gcb-pvt-key.pem"
  - "gcb-ca-root-cert.pem"
```

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)
```

- If you provide both IPv4 and IPv6 addresses in the above argument, the following configurations will be used:

```
- name: UCTC_CNTLR_FM_IP_CONFIG
  value: IPv4 or IPv6
- name: UCTC_CNTLR_FALLBACK_CONFIG
  value: True or 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 will be used.

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



#### Notes:

- Fallback configuration will be used during node registration phase only.
- Controller FM IP and FALLBACK configurations will be used only if you provide both IPv4 and IPv6.
- If you provide IPv4 alone, only IPv4 connections will be considered, and UCTC\_CNTLR\_FM\_IP\_CONFIG and UCTC\_CNTLR\_FALLBACK\_CONFIG will be disregarded. Similarly, if IPv6 alone is provided, only IPv6 connections will be considered.
- Default values will be used if you do not provide any options.

**Note**

In dual stack cluster, update the following fields in **uctc-cntlr-service** to deploy dual stack setup.

```
Spec:
ipFamilies:
- Ipv4
- IPv6
ipFamilyPolicy: PreferDualStack
```

- Provide **External IP** and **Kubernetes Cluster API URL** in the following section of the YAML file:

```
env:
- name: UCTC_CNTLRL_SERVICE_NAME
value: "GIGAMON_UCTC_CNTLRL_SERVICE"
- name: UCTC_CNTLRL_EXT_IP_DNS
value: "<external IPv4 or external IPv6 or external IPv4,IPv6 (both with comma-
separation notation) or DNS"
```

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)
# example4: gigamon.example.com
- name: K8S_CLUSTER_ENDPOINT
value: <K8s Cluster API URL>
```

Refer to the below example:

```
# example: https://10.10.10.13:6443
- name: FM_FQDN
value: www.fm.gigamon.com
```

- Update the namespace in the YAML file as required and run the following command:

```
kubectl create -f uctc-controller.yaml
```

Following the execution of the above command, when UCT-C Controller pod is created successfully, the output (sample) will be as below:

```
gigamon@controller-2:~$kubectl create -f uctc-controller.yaml
service/gigamon-uctc-cntlr-service created
deployment.apps/uctc-cntlr-v1 created
clusterrole.rbac.authorization.k8s.io/pods-list created
clusterrolebinding.rbac.authorization.k8s.io/pods-list created
```

## Deployment of UCT-C Tap

Follow these steps to deploy the UCT-C Tap:

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

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

After extracting the tar file, navigate to the YAML folder in the newly created **uctc-tap-  
<image version>-<build number>** folder and update the details given in the below steps.

- Feed the created **secret** in the below section of YAML file

```
imagePullSecrets:
  - name: <secret>
```

- Update the namespace in the below section of YAML file as required. This should be same as the namespace in which UCT-C controller is deployed.

```
- name: UCTC_CNTLRL_SVC_DNS
value: gigamon-uctc-cntlr-service.<namespace>.svc.cluster.local
```

- When UCT-C TAP gets both IPv4 and IPv6 from the above controller service DNS, the following configurations will be used:

```
- name: UCTC_TAP_IP_CONFIG
value: IPv4 or IPv6
- name: UCTC_TAP_FALLBACK_CONFIG
value: True or 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 will be used.

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



#### Notes:

- Fallback configuration will be considered during node registration only.
- If IPv4 alone is provided, only IPv4 connections will be considered, and UCTC\_TAP\_IP\_CONFIG and UCTC\_TAP\_FALLBACK\_CONFIG will be disregarded. Similarly, if IPv6 alone is provided, only IPv6 connections will be considered.
- Default values will be used if TAP gets dual IP's from the controller service DNS.

- Edit the following **volumeMounts** as per your container Runtime.

```
volumeMounts:
  - name: socket
    mountPath: /var/run/containerd/containerd.sock
volumes:
  - name: socket
    hostPath:
      Path: /var/run/containerd/containerd.sock
```

Below are the socket location for commonly used CRIs,

```
docker - /var/run/docker.sock
containerd - /var/run/containerd/containerd.sock
```



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

- o Run the following command for deploying UCT-C Tap:

```
kubectl create -f uctc-tap.yaml -n <namespace where UCT-C tap has to be deployed>
```

Following the execution of the above command, when UCT-C Tap pod is created successfully, the output (sample) will be as below:

```
gigamon@controller-2:~$ kubectl create -f uctc-tap.yaml -n uctc
daemonset.apps/gigamon-uctc created
```

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

Field Values	Description
Port: 443	The UCT-C Controller REST service port number.
Port: 42042	This port must be port 42042. Allows UCT-C to send statistics information to UCT-C Controller.
GigaVUE-FM IP	The IP address of the GigaVUE-FM with which your UCT-C is connected.
UCT-C-Cntrl REST SVC Port	The UCT-C Controller REST service port number. This must be opened on your GigaVUE-FM to allow inbound traffic to Kubernetes. This allows GigaVUE-FM to communicate with UCT-C Controller. Example: 8443 (configurable)
FM REST Svc Port	The GigaVUE-FM REST service port number. This must be opened on your Kubernetes to allow outbound traffic. This allows UCT-C Controller to communicate with GigaVUE-FM. Example: 443
Ports: containerPort: 443 containerPort: 42042	Two ports must be opened. The first container port must be the same as UCT-C-Cntrl REST SVC Port. The second container port must be port 42042. This allows UCT-C to send statistical data to UCT-C controller.
External LB balancer IP	The external load balancer IP/DNS value to allow GigaVUE-FM to communicate with UCT-C Controller within Kubernetes.
K8S cluster end-point	Kubernetes cluster end point for GigaVUE-FM to access the control plane. Example: https://<kubernetesapiserverurl>:6443

## Using Helm Charts

Helm Charts will be available for each and every release build in the Gigamon software portal. Download the respective UCT-C release build from the repository and untar the **.tgz** file. After you finish untarring the file, you can extract the Helm Charts (**uct-cntrl-  
<version>.tgz** and **uct-tap-<version>.tgz**) and further update the following fields before deployment.

**NOTE:** [Contact Technical Support](#) or [Contact Sales](#) for information on downloading the respective UCT-C build from the Gigamon software portal.

Refer to the following section for detailed information.

- [Deploy using Helm Chart](#)

**NOTE:** Support for two Helm Charts is deprecated from software version 6.7.00.

## Deploy using Helm Chart

You can deploy UCT-C Controller and TAPs using single Helm Chart. Follow the steps listed below to deploy the solution.

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

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

After extracting the tar file, navigate to the Helm folder in the newly created **uctc-  
<image version>-<build number>** folder and update the details given in the below steps.

2. Update the **imagePullSecrets, namespace, GigaVUE-FM IP, external load balancer IP** and **Kubernetes API URL** in the following section of the **values.yaml** file present in the UCT-C directory.

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

If you provide IPv4 alone and not IPv6, then GigaVUE-FM will consider only IPv4 address for communication.

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

If you provide IPv6 alone and not fm\_ip, then GigaVUE-FM will consider only IPv6 address for communication.

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

If both IPv4 and IPv6 are provided, **UCTC\_CNTLRL\_FM\_IP\_CONFIG** will be 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
```

- If you provide two IP's, IPv4 and IPv6 in the `fm_ip` argument, the following configurations will be used:

```
# 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 will be used.

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



#### Notes:

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

- 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
```

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

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

**NOTE:** Users can skip the above steps 1-5 and use the below command to directly deploy UCT-C Controller and TAPs using single Helm Chart.

```
helm install uctc -n uctc ./uctc --set namespace=uctc --set serviceAccount.name=test --
set imagePullSecrets[0].name=gigamon --
set uctcController.fm_ip=x.x.x.x --set uctcController.ext_load_balancer=x.x.x.x --set uctcController.k8s_
cluster_url=https://x.x.x.x: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
```

### 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** will be deployed, which will perform checks to ensure certain conditions are met for a successful

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

Follow the steps listed below, 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 failed) 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 UCT-C folder is present.

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

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

## Deploy UCT-C on Redhat Openshift Platform using Openshift UI

You can deploy the UCT-C Controller and TAPs in the Redhat Openshift Platform using Helm Charts. Refer to the following sections for detailed information.

- [Prerequisites](#)
- [Deployment of UCT-C Controller and TAPs](#)

### Prerequisites

- To deploy, you should have Developer access in Redhat Openshift Platform.
- To validate the deployment, you should have Administrator access.
- In Redhat Openshift Platform, you should create service account and add it to privileged access (use the below command). Use that service account in values.yaml which allows uctc-tap to come up as a privileged pod.

```
oc adm policy add-scc-to-user -z <service_account> <privileged_scc> -n uctc
```

## Deployment of UCT-C Controller and TAPs

To deploy UCT-C Controller and TAPs, follow the below- listed steps:

1. Log in to the Redhat Openshift online platform using your Redhat login credentials.
2. Switch to Developer access in the drop-down on the top of the page, navigate to the **Helm** section and click **Create > Helm Release**. Helm Charts screen appears.
3. Browse and select Gigamon from the **All items** search menu.
4. On the Gigamon-UCT-C landing page, click **Create. Create Helm Release** page appears.

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

5. To create Helm Release, enter or select the required information as described in the following table.

Section	Field	Description
	Release Name	Specify the Helm Release name.
	Chart Version	Select the appropriate release version from the drop-down menu. By default, the latest uploaded version of the release will be displayed.
	Configure via	Select between Form view and YAML view.
<b>Gigamon UCT-C Configuration</b>		
	imagePullSecrets	Specify the created secret name.
<b>uctcTap</b>	resources - crisocketvolume	Specify the socket location details.  <b>NOTE:</b> The socket location for commonly used CRIs are as follow: <b>docker</b> - /var/run/docker.sock <b>containerd</b> - /var/run/containerd/containerd.sock <b>cri-o</b> - /var/run/crio/crio.sock
	ingress	Specify the following details in the ingress section: <ul style="list-style-type: none"> <li>• <b>enabled</b> - Click the check box to enable</li> <li>• <b>className</b> - Specify the class name</li> <li>• <b>annotations</b> - Specify the annotations details (kubernetes.io/ingress.class and nginx.ingress.kubernetes.io/backend-protocol).</li> </ul>
	serviceAccount	Enable the Create option and specify the serviceAccount name.

Section	Field	Description
<b>uctcController</b>	resources	Specify the port value.
	certs	Specify the following details in the uctcController section: <ul style="list-style-type: none"> <li>• <b>ext_load_balancer</b> - The external load balancer IP/DNS value to allow GigaVUE-FM to communicate with UCT-C Controller within Kubernetes.</li> <li>• <b>k8s_cluster_url</b> - Kubernetes cluster end point for GigaVUE-FM to access the control plane. Example: https://&lt;kubernetesapiserverurl&gt;:6443</li> </ul>
	service Name	Specify the service label name. Example: uctc-cntrl-service
	image	Specify the following details in the image section: <ul style="list-style-type: none"> <li>• repository</li> <li>• pullPolicy</li> </ul>
	fm_ip	Update the fm_ip detail.
	<b>namespace</b>	Update the namespace detail.

6. Click **Create** to deploy the UCT-C solution.
7. To validate the deployment, switch to **Administrator** view and navigate to:
  - **DaemonSets** option to validate the UCT-C-Tap deployment.
  - **Deployment** option to validate the UCT-C-Controller deployment.

## Configure UCT-C Controller and TAP through GigaVUE-FM

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

- [Launch GigaVUE-FM](#)
- [Universal Cloud Tap - Container Inventory](#)
- [Create Monitoring Domain](#)
- [Create Source Selectors](#)
- [Create Tunnel Specifications](#)
- [Configure Traffic Policy](#)
- [Policies Landing Page](#)
- [Viewing Policy Configurations](#)
- [Traffic Policy Statistics](#)
- [Viewing Policy Statistics](#)

## Launch GigaVUE-FM

The recent GigaVUE-FM image files can be downloaded from the [Gigamon Customer Portal](#). After fetching the image, upload and launch GigaVUE-FM on the supported Cloud environment. For assistance, [Contact Technical Support](#) of Gigamon or refer to GigaVUE-FM Installation Guide and GigaVUE Fabric Management Guide for details on installing and launching GigaVUE-FM.

## Universal Cloud Tap - Container Inventory

In GigaVUE-FM, on the left navigation pane, 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 along with the connectivity status from GigaVUE-FM to Cluster.
Clusters	<p>Displays Kubernetes Clusters, along with UCT-C Controller information and total Nodes per Cluster. Also displays GigaVUE-FM to Controller connectivity and Heartbeats status.</p> <div data-bbox="646 995 1466 1152" style="border: 1px solid #ccc; padding: 5px;"> <p><b>NOTE:</b> You can delete a cluster that is not tagged to any Monitoring Domain from the Clusters page. Ensure to delete the cluster only after stopping UCT-C Controller and UCT-C TAP.</p> </div>
Nodes	<p>Displays the Nodes from all Kubernetes Cluster along with UCT-C TAP information and Total Pods per Node. UCT-C TAP status should be Connected 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.</p> <p>UCT-C TAP solution cannot tap traffic from Pods with Host Network Enabled.</p>
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.
Settings	Displays the general settings which include disconnected UCT-C 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, click the filter button on the left side of any of the above listed tabs. You can also create a new Monitoring Domain, edit, and delete the existing Monitoring Domains.

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

## Create Monitoring Domain

To create a monitoring domain in GigaVUE-FM:

1. In GigaVUE-FM, on the left navigation pane, go to **Inventory > CONTAINER > Universal Cloud Tap - Container > Monitoring Domains**.
2. In the **Monitoring Domains** page, click **New**. The **New Monitoring Domain** wizard appears.
3. Enter or select the required information as described in the following table.

Fields	Description
Monitoring Domain Name	Enter a name for the monitoring domain
Clusters	
Cluster Name	Enter a name for the cluster
API Server URL	Enter or select the URL of the API server
Option	
CA	Select the required CA name from the drop-down menu. <b>NOTE:</b> CA is required for deploying the policy with Secure Tunnels.



Click  to add another cluster and click  to remove an existing cluster.

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

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:

Button	Description
New	Use to create new Monitoring Domain
Actions	Provides the following options: <ul style="list-style-type: none"> <li><b>Edit</b>- Edit the monitoring domain</li> <li><b>Delete</b> - Delete the monitoring domain</li> </ul>
Refresh Inventory	Triggers Inventory Refresh on all Clusters in the Monitoring Domain

## Create Source Selectors

When setting up a traffic flow, it is important to define the selection criteria for the sources of traffic. Use the Source Selectors page to configure the sources of the traffic to be monitored.



### Notes:

- DefaultExclusion:** DefaultExclusion is a default source selector which will be applied to all policies. It cannot be deleted but can be modified. After modifying the DefaultExclusion source selector, policies need to be deployed again 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
- hostNetwork

By default pods in kube-system namespace, metallb-system namespace, and host network enabled pods 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.

Dashboard

Criteria 1 ▼

Object Property	Operator	Value	+	-
namespace ▼	equals ▼	kube-system	+	-

OR

Criteria 2 ▼





Object Property	Operator	Value	+	-
namespace ▼	equals ▼	metallb-system	+	-



Criteria 3 ▼

Object Property	Operator	Value	+	-
hostNetwork ▼	equals ▼	enabled ▼	+	-

To configure the Source Selectors:

1. Go to **Inventory > Resources> Source Selectors**.
2. On the **Source Selector Specifications** page, navigate to the **Container** tab and click **Create**. The **New Source Selector** wizard appears.
3. Enter or select the required information:

Field	Action
<b>Name</b>	Enter a name for the source
<b>Inclusion Criteria</b> You can select any one of the following options: <b>All Sources</b> - Select this option to acquire traffic from all namespaces and pods within the selected cluster(s). The volume of traffic may be large, depending on the size of the cluster(s). <b>Criteria1-</b> You must enter the following options:	
Object Property	Select an object property to filter the traffic source.
Operator	Select the operator.
Values	Enter the values for the filter. Values are case-sensitive.
On the Criteria, click  to add another Object and click  to remove an existing Object.	
<b>Exclusion Criteria</b> On the Criteria, click  to add another Object and click  to remove an existing Object.	
Object Property	Select an object property to filter the traffic source.
Operator	Select the operator.
Values	Enter the values for the filter. Values are case-sensitive.

On the Inclusion or Exclusion Criteria sections, click  to add another Criteria and click  to remove an existing Criteria.

4. Click **Save**.



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

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

## Create Tunnel Specifications

A tunnel of type L2GRE, VXLAN, or TLS-PCAPNG can be created. The tunnel is an egress tunnel. For more information on how to create a tunnel of type TLS-PCAPNG, refer to [Secure Tunnels Secure Tunnels . Secure Tunnels](#)

To configure the tunnels:

1. Go to **Inventory > Resources > Tunnel Specifications**.
2. On the **Tunnel Specifications** page, navigate to the **Container** tab and click **Create**. The **Create Tunnel Specification** wizard appears.
3. Enter or select the following information:

Field	Description
Name	The name of the tunnel endpoint.  <b>NOTE:</b> Do not enter space or Invalid characters.
Tunnel Type	Select L2GRE, VXLAN, or TLS-PCAPNG tunnel type to create a tunnel.
Destination IP Address	Enter the IP address of the destination endpoint.
Key <b>(Applicable when the selected tunnel type is L2GRE)</b>	Enter a value for the tunnel key.
VXLAN Network Identifier <b>(Applicable when the selected tunnel type is VXLAN)</b>	Enter the identifier key for VXLAN network.
Destination Port <b>(Applicable when the selected tunnel type is VXLAN or TLS-PCAPNG)</b>	Specify the destination port value. Enter a value between 1 and 65535.

4. Click **Save** to save the configuration.

## 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 a UCT-C Traffic Policy in GigaVUE-FM, follow the below steps:

1. From the GigaVUE-FM left navigation pane, go to **Traffic > CONTAINER > Universal Cloud Tap - Container**. The **Policies** page appears.
2. In the **Policies** page, click **New**. The Create Policy wizard appears.

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



3. In the **General** tab, enter or select the required information as described in the following table:

Fields	Description
Policy Name	Enter a name for the Traffic Policy. The name must be unique.
Monitoring Domain	Select an existing monitoring domain. To create a new monitoring domain, refer to <a href="#">Create Monitoring Domain</a> section.
Clusters	Select the required cluster from the drop-down menu.
Precription Policy	Click the radio button <b>Yes</b> , to enable the Precription rules for the policy. Click radio button <b>No</b> to enable the Mirroring. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p><b>NOTE:</b> Once the policy is deployed, you cannot change the Precription Policy setting.</p> </div>

4. Click **Next** to switch to the **Source Selectors** tab, select an existing source selector and click **Add** or select **Create New** to create a new source selector, refer to [Create Source Selectors](#) section for detailed information. You can configure a maximum of eight source selectors per policy.
5. In the **Source Selectors** page, click  to expand the **Default Exclusion** section. DefaultExclusion Source Selector is applied automatically for all policies.


**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.

6. Click **Next** to switch to the **Rules** tab, enter or select the required information for the **Ingress Rules** and the **Egress Rules** as described in the following table. You must select CA in the Monitoring Domain page to use secure tunnel in rules:

Fields	Description
Tunnel Specifications	<p>Select an existing tunnel or select <b>Create New</b> to create a new tunnel, refer to <a href="#">Create Tunnel Specifications</a> section for detailed information.</p> <p>For Precryption, only one Tunnel Specification field will be displayed at the top for all the rules. For Mirroring, Tunnel Specification can be configured for every individual rule.</p>
<p><b>Rules</b></p> <p>On the Ingress or Egress rules, click  to add another rule and click  to remove an existing rule. You must select CA in the Monitoring Domain page to use secure tunnel in rules.</p>	
Rule Name	<p>Enter a name for the rule. Rule name should always be unique within a policy.</p> <p><b>NOTE:</b> Rule names ending with <code>_I</code>, <code>_E</code>, <code>_RI</code>, <code>_RE</code> are not recommended as the names are invalid in policy rules. Rule names like <code>passall</code>, <code>ingress-passall</code>, and <code>egress-passall</code> are restricted.</p>
Enable	<p>Select <b>On</b> to enable the passall rule or select <b>Off</b> to disable the passall rule. Refer to <a href="#">Enable Selective Precryption</a> to add the filters when you choose to disable the passall rule.</p>
Action	<p>Select Pass to allow the packets or select Drop to block the packets based on the filters.</p>
Direction	<p>Select any one of the following directions:</p> <ul style="list-style-type: none"> <li>• Bi-directional - Taps the traffic in both directions. Each bidirectional rule will add 2 ingress rules and 2 egress rules</li> </ul> <p><b>NOTE:</b> When you apply filters to two pods on the same worker node to capture traffic in both directions, only one copy of the packet will be tunneled out for each packet traveling from one pod to the other.</p> <ul style="list-style-type: none"> <li>• Ingress- Taps the ingress traffic</li> <li>• Egress - Taps the egress traffic</li> <li>• Ingress Pass All - Taps all the ingress traffic</li> <li>• Egress Pass All - Taps all the egress traffic</li> </ul> <p><b>NOTE:</b> The maximum number of rules supported per direction is 32.</p>
Priority	<p>Enter a priority value to specify the order of rule execution on the selected Pod.</p> <p>Unique Priority is enforced in a policy within ingress and egress space. Bi-directional 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.</p>

## Enable Selective Precryption

If you wish to use selective Precryption follow the steps given below:

- a. Disable the **Enable** toggle button to turn off the default passall rule.
- b. Click  to add another rule.

c. Enter the following details as mentioned in the below table:

Fields	Description
Rule Name	Enter a name for the rule.
Action	<p>Choose any one of the following options:</p> <ul style="list-style-type: none"> <li>• Pass — Passes the traffic.</li> <li>• Drop — Drops the traffic.</li> </ul> <p><b>NOTE:</b> In the absence of a Precryption rule, traffic is implicitly allowed. However, once rules are defined, they include an implicit pass all rule. Should the traffic not conform to any of the specified rules, it will be passed.</p>
Direction	<p>Choose any one of the following options:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Ingress — Filters the traffic that flows in.</li> <li>• Egress — Filters the traffic that flows out.</li> </ul>
Priority	Select the value of the priority based on which the rules must be prioritized 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 can be used for setting a rule with the least priority. Drop rules are added based on the priority and, then pass rules are added.
Filters	
Filter Type	<p>Select the <b>Filter Type</b> from the following options:</p> <ul style="list-style-type: none"> <li>• L3</li> <li>• L4</li> </ul>
L3:	
Filter Name	<p>Select the <b>Filter Name</b> from the following options:</p> <ul style="list-style-type: none"> <li>• IPv4 Source</li> <li>• IPv4 Destination</li> <li>• IPv6 Source</li> <li>• IPv6 Destination</li> <li>• Protocol - It is common for both IPv4 and IPv6</li> </ul>
Value	<p>Enter or Select the Value based on the selected Filter Name.</p> <p><b>NOTE:</b> When using <b>Protocol</b> as the <b>Filter Name</b>, select <b>TCP</b> from the drop-down menu.</p>
L4:	



Fields	Description
Filter Name	Select the <b>Filter Name</b> from the following options: <ul style="list-style-type: none"> <li>• Source Port</li> <li>• Destination Port</li> </ul>
Filter Relation	Select the <b>Filter Relation</b> from any one of the following options: <ul style="list-style-type: none"> <li>• Not Equal to</li> <li>• Equal to</li> </ul>
Value	Enter the source or destination port value.


7. Click **Next** to switch to the **Validate** tab, click **Save** to the policy or click **Deploy**, and the selected traffic policy rules get deployed to the required UCT-C TAPs present on the nodes corresponding to the source pods selected for monitoring.

## Policies Landing Page

In the **Policies** page, you can view various details related to a policy such as **Policy Name**, **Monitoring Domain**, **Cluster**, **Deployment Status**, and **Statistics** etc., The fields and the description of the field names are given in the following table:

Table 1: Policies Landing Page

Fields	Description
Policy	Name of the Policy
Monitoring Domain	Monitoring Domain associated with the Policy
Cluster	The cluster name associated with the policy
Deployment Status	Indicates the policy deployment status: <ul style="list-style-type: none"> <li>• Deployed</li> <li>• Not Deployed</li> <li>• Undeploying</li> <li>• UndeploymentFailure</li> </ul>
Statistics	Displays the statistics of the policy

**NOTE:** Click the gear icon  to add or remove column or columns as per your requirement.

Use the following buttons on the Policies screen to manage your Policy:

Button	Description
New	Use to create a new policy
Actions	Provides the following options: <ul style="list-style-type: none"> <li>• Edit</li> <li>• Delete</li> <li>• Deploy</li> <li>• Undeploy</li> </ul>

To view the Deployment status, click the **Deployment** status link in the Deployment Status column of a policy. The Deployment status appears on the bottom of the **Policies** page. You can view the following fields along with the policy name:

- Pod Name
- Namespace
- Rules
- Cluster
- Node

Click **Filter** to filter the details like Cluster, Node, Namespace, Pod Name, and Pod Status in the Deployment Status Wizard.

## Viewing Policy Configurations

To view the Policy Configurations of the traffic policy configured in the GigaVUE-FM, click the 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 Rules or Precryption Rules](#)

You can scroll each of the tables to view more columns. The fields and description for the tab that appears when you click the tabs are described in the topics respectively.

### Source Specifications

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

The fields and descriptions of the **Source Specifications** tab are described in the following table:

Table 2: Source Specifications

Tab- Source Specifications	Field	Description
<b>Source Selector</b>		
	Name	Specifies the name of the Source selector.
<b>Inclusion Criteria</b>		
	Criteria Name	Specifies the inclusion criteria for the source selector. Pod that matches the inclusion criteria will be selected as source for the given traffic policy.
	Property	Specifies the property for the attributes in the criteria. The available properties are: <ul style="list-style-type: none"> <li>• namespace</li> <li>• servicename</li> <li>• serviceip</li> <li>• podname</li> <li>• podip</li> <li>• podlables</li> <li>• nodename</li> <li>• nodepodcidr</li> </ul>
	Operator	Specifies the operator used in the criteria.
	Value	Specifies the value for the attributes in the criteria.
<b>Exclusion Criteria</b>		
	Criteria Name	Specifies the exclusion criteria for the source selector. Pod that matches the exclusion criteria will be excluded from the source for the given traffic policy.
	Property	Specifies the property in the exclusion criteria based on which the pod associated with the source is excluded.
	Operator	Specifies the operator involved in the exclusion 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 Rules or Precryption Rules

The fields and descriptions of the **Mirroring Rules or Precryption Rules** tab are described in the following table:

Table 3: Mirroring Rules or Precryption Rules

Tab-Rules	Field	Description
<b>Mirroring Rules or Precryption Rules</b>		
	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 the direction of the flow of traffic is ingress, egress, or in both direction.
<b>Filter</b>		
	Type	Specifies the filter type.
	Filter	Specifies the name for the filter.
	Value	Specifies the value of the filter.

## Traffic Policy Statistics

Traffic Policy Statistics in GigaVUE-FM provides the visibility of the policies within a Monitoring Domain and displays the information of the policies and its rules statistics in the dashboard.

Rules are configured in the UCT-C to either forward the traffic to a Tunnel or drop the flow of the traffic.

The activities of the rules are reflected by the statistics counters. The statistics counters show how the policy statistics are directly co-related to the policy and its rules being configured through the GigaVUE-FM.

## Viewing Policy Statistics

To view the statistics of the traffic policy configured in the GigaVUE-FM, click the **View** status link in the **Statistics** column of a policy. The Policy Statistics and the Mirroring Statistics or Precryption Statistics appears on the bottom of the **Policies** page:

Table 4: 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

Table 5: Mirroring or Precryption Statistics

Fields	Description
Rule Name	Name of the individual rule.
Ingress packets	Total aggregate value of the ingress packets associated with the rule
Egress packets	Total aggregate value of the egress packets associated with the rule
Ingress Bytes	Total aggregate value of the ingress bytes associated with the rule
Egress Bytes	Total aggregate value of the egress bytes associated with the rule
Ingress Errors	Total aggregate value of the ingress errors associated with the rule
Egress Errors	Total aggregate value of the egress errors associated with the rule
Ingress Dropped	Total aggregate value of the ingress packets dropped associated with the rule
Egress Dropped	Total aggregate value of the egress packets dropped associated with the rule

## Precryption™

**License:** Requires **SecureVUE Plus** license.

Gigamon Precryption™ technology<sup>1</sup> redefines security for virtual, cloud, and containerized applications, delivering plain text visibility of encrypted communications to the full security stack without the traditional cost and complexity of decryption.

This section explains:

- [How Gigamon Precryption Technology Works](#)

---

<sup>1</sup> **Disclaimer:** The Precryption feature allows users to acquire traffic after it has been decrypted. This traffic can be acquired from both virtual machine (VM) and container-based solutions, and is then sent to the V Series product for further processing. The Precryption feature provides an option to use encrypted tunnels for communication between the acquisition (via UCT-C or UCT-V) of unencrypted traffic and the traffic processing (at the V Series) which will better safeguard the traffic while in transit. However, if a user does not use the option for encrypted tunnels for communication, decrypted traffic will remain unencrypted while in transit between the point of acquisition and processing. Please note that this information is subject to change, and we encourage you to stay updated on any modifications or improvements made to this feature. By using this feature, you acknowledge and accept the current limitations and potential risks associated with the transmission of decrypted traffic.

- [Why Gigamon Precryption](#)
- [Key Features](#)
- [Key Benefits](#)
- [Precryption Technology on Single Node](#)
- [Precryption Technology on Multi-Node](#)
- [Supported Platforms](#)
- [Prerequisites](#)

## How Gigamon Precryption Technology Works

Precryption technology leverages native Linux functionality to tap, or copy, communications between the application and the encryption library, such as OpenSSL.



In this way, Precryption captures network traffic in plaintext, either before it has been encrypted or after it has been decrypted. Precryption functionality doesn't interfere with the message's actual encryption or transmission across the network. There's no proxy, retransmissions, or break-and-inspect. Instead, this plaintext copy is forwarded to the Gigamon Deep Observability Pipeline for further optimization, transformation, replication, and tool delivery.

Precryption technology is built on GigaVUE® Universal Cloud Tap (UCT) and works across hybrid and multi-cloud environments, including on-prem and virtual platforms. As a bonus, UCT with Precryption technology runs independently of the application and doesn't have to be baked into the application development life cycle.

## 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. It provides East-West visibility into virtual, cloud, and container platforms. It delivers unobscured visibility into all encryption types, including TLS 1.3, without managing and maintaining decryption keys. IT organizations can now manage compliance, keep private communications private, architect the necessary foundation for Zero Trust, and boost 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:

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

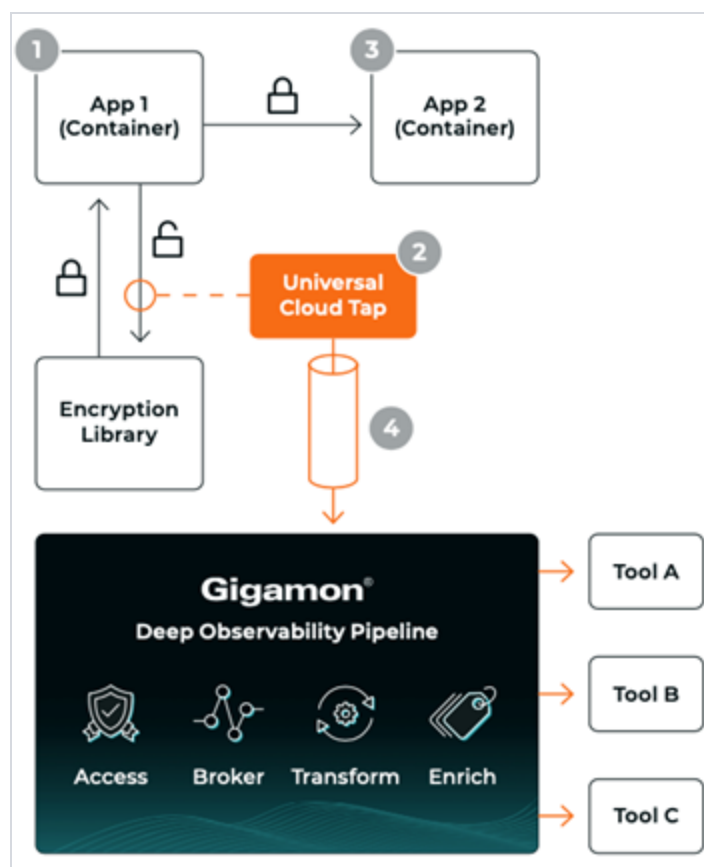
## How Gigamon Precryption Technology Works

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](#)

## Precription Technology on Single Node

1. When any application needs to encrypt a message, it uses an encryption library, such as OpenSSL, to perform the actual encryption.
2. GigaVUE Universal Cloud Tap (UCT), enabled with Precription technology, gets a copy of this message before it's 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. Gigamon optimizes, transforms, and delivers data to tools without further decryption.

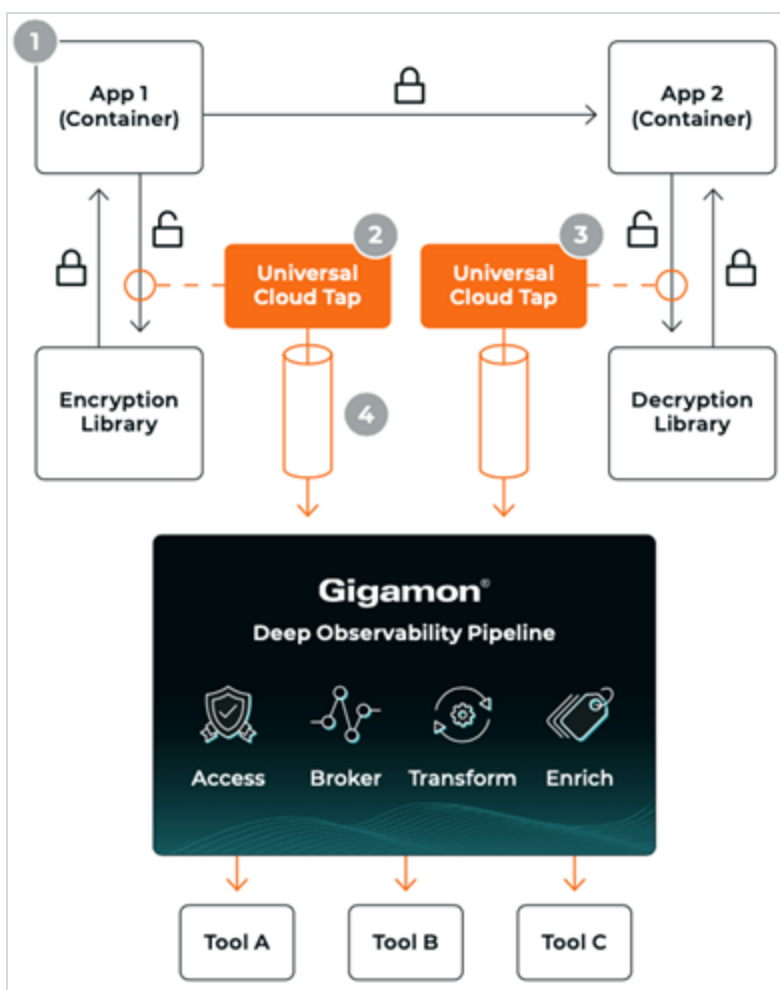


## Precription Technology on Multi-Node

1. When any application needs to encrypt a message, it uses an encryption library, such as OpenSSL, to perform the actual encryption.
2. GigaVUE Universal Cloud Tap (UCT), enabled with Precription, gets a copy of this message before it's encrypted on the network.
3. Optionally, GigaVUE UCT enabled with Precription can also acquire a copy of the message from the server end after the decryption.



- GigaVUE UCT creates packet headers as needed, encapsulates them in a tunnel, and forwards them to V Series in the deep observability pipeline. There, they are further enriched, transformed, and delivered to tools without further decryption.



## Supported Platforms

**VM environments:** Precryption™ is supported on the following VM platforms where UCT-V is supported:

Platform Type	Platform
Public Cloud	<ul style="list-style-type: none"> <li>AWS</li> <li>Azure</li> <li>GCP (via Third Party Orchestration)</li> </ul>
Private Cloud	<ul style="list-style-type: none"> <li>OpenStack</li> <li>VMware ESXi (via Third Party Orchestration only)</li> <li>VMware NSX-T (via Third Party Orchestration only)</li> </ul>

**Container environments:** Precryption™ is supported on the following container platforms where UCT-C is supported:

Platform Type	Platform
Public Cloud	<ul style="list-style-type: none"> <li>EKS</li> <li>AKS</li> </ul>
Private Cloud	<ul style="list-style-type: none"> <li>OpenShift</li> <li>Native Kubernetes (VMware)</li> </ul>

## Prerequisites

### Deployment Prerequisites

- 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 precrypted packets from the worker pods whenever libssl calls are made from the worker pods.
- For GigaVUE-FM, you must add port 5671 in the security group to capture the statistics.
- Port 9900 should be enabled in security group settings on the UCT-V controller to receive the statistics information from UCT-V.
- For UCT-C, you must add port 42042 and port 5671 to the security group.

### License Prerequisite

- Precryption™ requires a SecureVUE Plus license.

### Supported Kernel Version

Precryption is supported for Kernel Version 5.4 and above for 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)
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	ubuntu19.10
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

Kernel-Version	Operating System
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

### Note

- See the [Configure Precryption in UCT-V](#) section for details on how to enable Precryption™ in VM environments.
- See the [Configure Precryption in UCT-C](#) section for details on how to enable Precryption™ in container environments.
- See how [Secure Tunnels](#) feature can enable secure delivery of precrypted data.

## Configure Precryption in UCT-C

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

### Rules and Notes

The following are the memory limits to be applied to 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).
- When you deploy secure tunnels, it requires additional (16 \*64 MB) memory. Hence, the total memory that you must allocate for the TAP is 1 GB.
- You can always configure the memory allocation using PRECRYPTION\_RING\_BUFFER\_MEMORY\_MB in YAML file.
- Protocol version IPv4 and IPv6 are supported.
- If you wish to use IPv6 tunnels, your GigaVUE-FM and the fabric components version must 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 below:

1. Go to **Traffic > CONTAINER > Universal Cloud Tap - Container**.
2. On the Policies page that appears, click **New**.

3. In the **General** tab, enter or select the required information as described in the following table:

Fields	Description
Policy Name	Enter a name for the Traffic Policy. The name must be unique.
Monitoring Domain	Select an existing monitoring domain. To create a new monitoring domain, refer to <a href="#">Configure Precryption in UCT-C</a> section.
Clusters	Select the required cluster from the drop-down menu.
Precryption Policy	Click the radio button <b>Yes</b> , to enable the Precryption rules for the policy. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>NOTE:</b> Once the policy is deployed, you cannot change the Precryption Policy setting.</p> </div>

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-V level. This filtering is done based on L3/L4 5 tuple information (5-tuple filtering) running on the containers.

Refer to [Enable Selective Precryption](#) for more detailed information on how to configure Selective Precryption when configuring the **Rules**.

## Secure Tunnels

Secure Tunnel securely transfers the cloud-captured packets on UCT-V and UCT-C to a GigaVUE V Series Node or Tool (only in the case of UCT-C). The data from UCT-V and UCT-C are encapsulated in PCAPng format, and the encrypted data is sent over a TLS connection to a GigaVUE V Series Node.

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

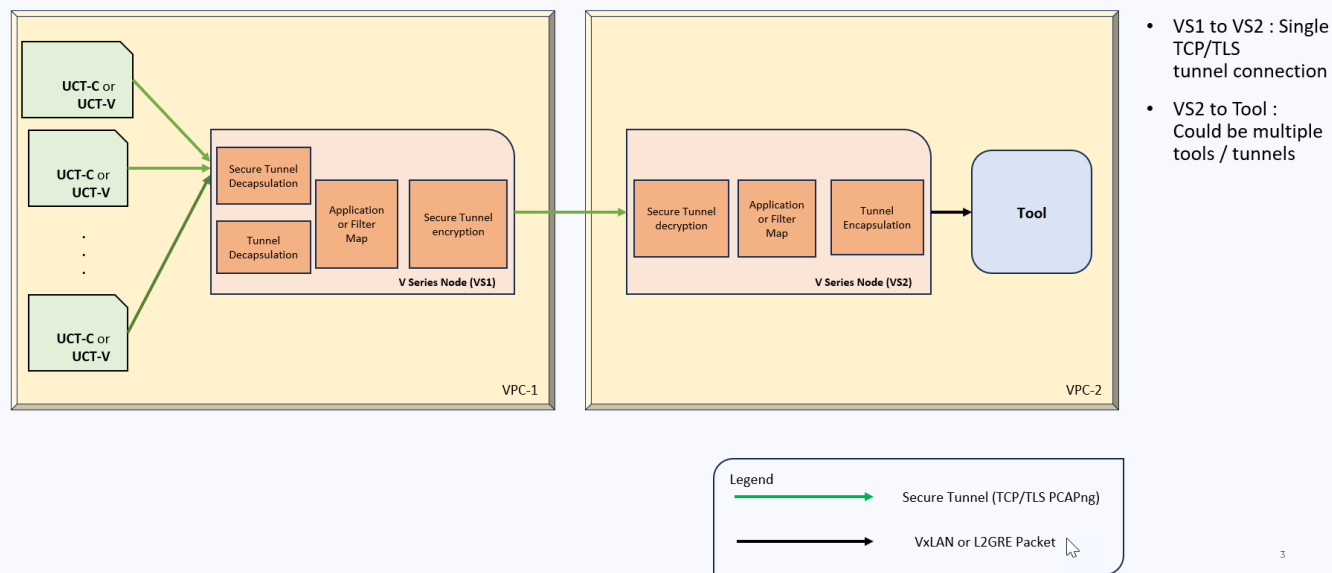
In the case of GigaVUE V Series Node to GigaVUE V Series node, the traffic from the GigaVUE V Series Node 1 is encapsulated using PCAPng format and transported to GigaVUE V Series Node 2, where the traffic is decapped. The secure tunnels between a V Series Node and a V Series Node have multiple use cases.

The GigaVUE V Series Node decapsulates and processes the packet as per the configuration. The decapsulated packet can be sent 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.

## Secure Tunnel Use Case

Tool in remote Virtual Private Cloud (VPC) – Single V Series Node



## Supported Platforms

Secure Tunnels is 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 Tunnel in UCT-C](#).

## Configure Secure Tunnel in UCT-C

Secure tunnel can be configured on:

- [Preencrypted Traffic](#)
- [Mirrored Traffic](#)

## Preencrypted Traffic

You can send the preencrypted traffic through secure tunnel. When secure tunnel for Preencryption 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 Preencryption packets, then two TLS secure tunnel sessions are created.

It is recommended to always enable secure tunnels for preencrypted traffic to securely transfer the sensitive information.

For more information about PCAPng, refer to [PCAPng Application](#).

## 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 Tunnel from UCT Container to GigaVUE V Series Node

To configure a secure tunnel in a UCT 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:

S. No	Task	Refer to						
1	Upload a Custom Certificate	<p>You must upload a CA to UCT Container for establishing a connection with the GigaVUE Cloud Suite V Series node.</p> <p>To upload the CA using GigaVUE-FM follow the steps given below:</p> <ol style="list-style-type: none"> <li>1. Go to <b>Inventory &gt; Resources &gt; Security &gt; CA List</b>.</li> <li>2. Click <b>New</b>, to add a new Certificate Authority. The <b>Add Certificate Authority</b> page appears.</li> <li>3. Enter or select the following information. <table border="1" data-bbox="716 514 1474 711"> <thead> <tr> <th>Field</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Alias</td> <td>Alias name of the CA.</td> </tr> <tr> <td>File Upload</td> <td>Choose the certificate from the desired location.</td> </tr> </tbody> </table> </li> <li>4. Click <b>Save</b>.</li> </ol> <p>For more information, refer to the section <a href="#">Adding Certificate Authority</a></p>	Field	Action	Alias	Alias name of the CA.	File Upload	Choose the certificate from the desired location.
Field	Action							
Alias	Alias name of the CA.							
File Upload	Choose the certificate from the desired location.							
2.	Upload a SSL Key	You must add a SSL key to GigaVUE Cloud Suite V Series node. To add SSL Key, follow the steps in the section <a href="#">SSL Decrypt</a> .						
3	Selecting the SSL Key when you create a monitoring domain and configure the fabric components in GigaVUE-FM.	To select the SSL Key follow the steps in the section <a href="#">SSL Decrypt</a> .						
4	Selecting the CA certificate when you create a monitoring domain and configuring the fabric components in GigaVUE-FM.	You should select the added Certificate Authority (CA) in UCT Container. To select the CA certificate, follow the steps in the section <a href="#">Create Monitoring Domain</a> .						
5.	Creating and adding the secure tunnel when you configure the traffic policy.	To create and add the secure tunnel while configuring in , in UCT Container refer to the <a href="#">Configure Traffic Policy</a>						

## Adding Certificate Authority

This section describes how to add Certificate Authority in GigaVUE-FM.

The Certificate Authority (CA) List page allows you to add the root CA for the devices.

To upload the CA using GigaVUE-FM follow the steps given below:

1. Go to **Inventory > Resources > Security > CA List**.
2. Click **Add**, to add a new Custom Authority. The **Add Certificate Authority** page appears.
3. Enter or select the following information.

Field	Action
<b>Alias</b>	Alias name of the CA.
<b>Certificate Authority</b>	Use any of the following option to enter the Certificate Authority
<b>Copy and Paste</b>	
Certificate	Enter the certificate.
<b>Install from URL</b>	
Path	Enter the URL in the following format: <protocol>://<username>@<hostname/IP address>/<file path>/<file name>
Password	Enter the password
<b>Install from Local Directory</b>	
File Name	Click <b>Choose File</b> button and choose the certificate from the desired location.

4. Click **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**, and the **Settings** page will appear with the existing General settings.
2. On the **General** section, click **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.
  - 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. Click **Save** to save the updates made on the General Settings

## UCT-C Log Level Settings

In GigaVUE-FM, you can control the level of logs created at each individual UCT-C TAP for troubleshooting. The default UCT-C log file name format is **uctc\_tap.log**.

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

1. In GigaVUE-FM, navigate to **Inventory > CONTAINER > Universal Cloud Tap - Container > Nodes**, the **Nodes** page appears with the existing information about the node, cluster, pod, and UCT-C details.
2. On the **UCT-C TAP** section, on any Monitoring Domain, click on the UCT-C TAP link, which is in connected status. The Universal Cloud Tap - Container Settings quick view appears.
3. Edit the following required UCT-C log values in the **LOGGING** section.
  - a. 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.
  - b. Enter a value for the number of lines in the UCT-C log file size field.
4. 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 6.9:** 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. To upgrade the GigaVUE-FM in respective cloud platforms, refer to [GigaVUE-FM Installation and Upgrade guide](#).

**NOTE:** GigaVUE-FM 6.9 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 6.9):** When upgrading GigaVUE-FM from version 6.8 to 6.9, all the UCT-C configurations will be retained in GigaVUE-FM allowing for a direct upgrade from version 6.8 to 6.9. GigaVUE-FM 6.9 is compatible with UCT-C versions 6.8 and 6.9.
3. **Upgrade to UCT-C 6.9:** Before upgrading UCT-C from earlier versions less than 6.8, you must delete the older versions and deploy UCT-C 6.9 version. To deploy UCT-C, refer to [Steps to Delete and Redeploy the UCT-C Solution](#). Post UCT-C upgrade to 6.9, verify that the controller version is 6.9.00.
4. **Upgrade to UCT-C (6.8 to 6.9):** When upgrading UCT-C from version 6.8 to 6.9, you should upgrade the UCT-C Controller first and then upgrade UCT-C TAP.

**NOTE:** 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.

## 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.

# Steps to Delete and Redeploy the UCT-C Solution

## Using YAML files

1. To delete the UCT-C Controller or TAP using YAML files, run the below-listed commands:

- a. To delete UCT-C Controller

```
kubectl delete -f uctc-controller.yaml
```

- b. To delete UCT-C TAP

```
kubectl delete -f uctc-tap.yaml -n <namespace>
```

2. To redeploy the UCT-C Controller or TAP using YAML files, download the UCT-C Controller and TAP yaml files, edit the values (imagePullSecrets, FM IP address, External IP, Kubernetes API URL, VolumeMounts) in those yaml files, and run the below-listed commands:

- a. To install UCT-C Controller

```
kubectl create -f uctc-controller.yaml
```

- b. To install UCT-C TAP

```
kubectl create -f uctc-tap.yaml -n <namespace>
```

For more details on deployment of UCT-C Controller and TAPs using YAML files, refer to [YAML Files](#).

## 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>
```

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

## Debuggability and Troubleshooting

- For analyzing the issues, log into the UCT-C Controller /UCT-C TAP pod using the following command and verify the logs present in pod-data folder.
 

```
kubectl -it exec <<pod name>> -n <<namespace>> -- bash
```
- 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.

# 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 VUE 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 6.9 Hardware and Software Guides
<p><b>DID YOU KNOW?</b> 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 <b>Edit &gt; Advanced Search</b> from the menu. This opens an interface that allows you to select a directory and search across all PDFs in a folder.</p>
<p><b>Hardware</b></p> <p>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</p>
<b>GigaVUE-HC1 Hardware Installation Guide</b>
<b>GigaVUE-HC3 Hardware Installation Guide</b>
<b>GigaVUE-HC1-Plus Hardware Installation Guide</b>
<b>GigaVUE-HCT Hardware Installation Guide</b>
<b>GigaVUE-TA25 Hardware Installation Guide</b>
<b>GigaVUE-TA25E Hardware Installation Guide</b>
<b>GigaVUE-TA100 Hardware Installation Guide</b>

<b>GigaVUE Cloud Suite 6.9 Hardware and Software Guides</b>	
<b>GigaVUE-TA200 Hardware Installation Guide</b>	
<b>GigaVUE-TA200E Hardware Installation Guide</b>	
<b>GigaVUE-TA400 Hardware Installation Guide</b>	
<b>GigaVUE-OS Installation Guide for DELL S4112F-ON</b>	
<b>G-TAP A Series 2 Installation Guide</b>	
<b>GigaVUE M Series Hardware Installation Guide</b>	
<b>GigaVUE-FM Hardware Appliances Guide</b>	
<b>Software Installation and Upgrade Guides</b>	
<b>GigaVUE-FM Installation, Migration, and Upgrade Guide</b>	
<b>GigaVUE-OS Upgrade Guide</b>	
<b>GigaVUE V Series Migration Guide</b>	
<b>Fabric Management and Administration Guides</b>	
<b>GigaVUE Administration Guide</b>	covers both GigaVUE-OS and GigaVUE-FM
<b>GigaVUE Fabric Management Guide</b>	how to install, deploy, and operate GigaVUE-FM; how to configure GigaSMART operations; covers both GigaVUE-FM and GigaVUE-OS features
<b>Cloud Guides</b>	
how to configure the GigaVUE Cloud Suite components and set up traffic monitoring sessions for the cloud platforms	
<b>GigaVUE V Series Applications Guide</b>	
<b>GigaVUE Cloud Suite Deployment Guide - AWS</b>	
<b>GigaVUE Cloud Suite Deployment Guide - Azure</b>	
<b>GigaVUE Cloud Suite Deployment Guide - OpenStack</b>	
<b>GigaVUE Cloud Suite Deployment Guide - Nutanix</b>	
<b>GigaVUE Cloud Suite Deployment Guide - VMware (ESXi)</b>	
<b>GigaVUE Cloud Suite Deployment Guide - VMware (NSX-T)</b>	
<b>GigaVUE Cloud Suite Deployment Guide - Third Party Orchestration</b>	
<b>GigaVUE Cloud Suite Deployment Guide Universal Cloud Tap - Container</b>	
<b>Gigamon Containerized Broker Deployment Guide</b>	

<b>GigaVUE Cloud Suite 6.9 Hardware and Software Guides</b>	
<b>GigaVUE Cloud Suite Deployment Guide - AWS Secret Regions</b>	
<b>GigaVUE Cloud Suite Deployment Guide - Azure Secret Regions</b>	
<b>Reference Guides</b>	
<b>GigaVUE-OS CLI Reference Guide</b>	library of GigaVUE-OS CLI (Command Line Interface) commands used to configure and operate GigaVUE HC Series and GigaVUE TA Series devices
<b>GigaVUE-OS Security Hardening Guide</b>	
<b>GigaVUE Firewall and Security Guide</b>	
<b>GigaVUE Licensing Guide</b>	
<b>GigaVUE-OS Cabling Quick Reference Guide</b>	guidelines for the different types of cables used to connect Gigamon devices
<b>GigaVUE-OS Compatibility and Interoperability Matrix</b>	compatibility information and interoperability requirements for Gigamon devices
<b>GigaVUE-FM REST API Reference in GigaVUE-FM User's Guide</b>	samples uses of the GigaVUE-FM Application Program Interfaces (APIs)
<b>Factory Reset Guidelines for GigaVUE-FM and GigaVUE-OS Devices</b>	Sanitization guidelines for GigaVUE Fabric Management Guide and GigaVUE-OS devices.
<b>Release Notes</b>	
<b>GigaVUE-OS, GigaVUE-FM, GigaVUE-VM, G-TAP A Series, and GigaVUE Cloud Suite Release Notes</b>	new features, resolved issues, and known issues in this release ; important notes regarding installing and upgrading to this release
	<b>NOTE:</b> Release Notes are not included in the online documentation.
	<b>NOTE:</b> Registered Customers can log in to <a href="#">My Gigamon</a> to download the Software and Release Notes from the Software and Docs page on to <a href="#">My Gigamon</a> . Refer to <a href="#">How to Download Software and Release Notes from My Gigamon</a> .
<b>In-Product Help</b>	
<b>GigaVUE-FM Online Help</b>	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](mailto: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		
<b>About You</b>	<b>Your Name</b>	
	<b>Your Role</b>	
	<b>Your Company</b>	
<b>For Online Topics</b>	<b>Online doc link</b>	<i>(URL for where the issue is)</i>
	<b>Topic Heading</b>	<i>(if it's a long topic, please provide the heading of the section where the issue is)</i>

<b>For PDF Topics</b>	<b>Document Title</b>	<i>(shown on the cover page or in page header )</i>
	<b>Product Version</b>	<i>(shown on the cover page)</i>
	<b>Document Version</b>	<i>(shown on the cover page)</i>
	<b>Chapter Heading</b>	<i>(shown in footer)</i>
	<b>PDF page #</b>	<i>(shown in footer)</i>
<b>How can we improve?</b>	<b>Describe the issue</b>	<i>Describe the error or issue in the documentation. (If it helps, attach an image to show the issue.)</i>
	<b>How can we improve the content?</b> <b>Be as specific as possible.</b>	
	<b>Any other comments?</b>	

## 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](mailto: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](mailto:inside.sales@gigamon.com)

**Partners:** [www.gigamon.com/partners.html](http://www.gigamon.com/partners.html)

## Premium Support

Email Gigamon at [inside.sales@gigamon.com](mailto: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](http://community.gigamon.com)

**Questions?** Contact our Community team at [community@gigamon.com](mailto: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)