Micro Focus Transformation Hub

Software Version: 3.2.0

Administrator's Guide

Document Release Date: April 30, 2020
Software Release Date: April 30, 2020

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, "commercial computer software" is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation ("FAR") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("DOD"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement ("DFARS") and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.microfocus.com/t5/Transformation-Hub-Documentation/tkb-p/Transformation_Hub

Contents

Chapter 1: Transformation Hub

Transformation Hub is the high-performance message bus for security, network, flows, application, and other events. It can queue, transform, and route security events to other ArcSight or third-party software. This Kafka-based platform allows ArcSight components like Logger, ESM, and Investigate to receive the event stream, while smoothing event spikes, and functioning as an extended cache.

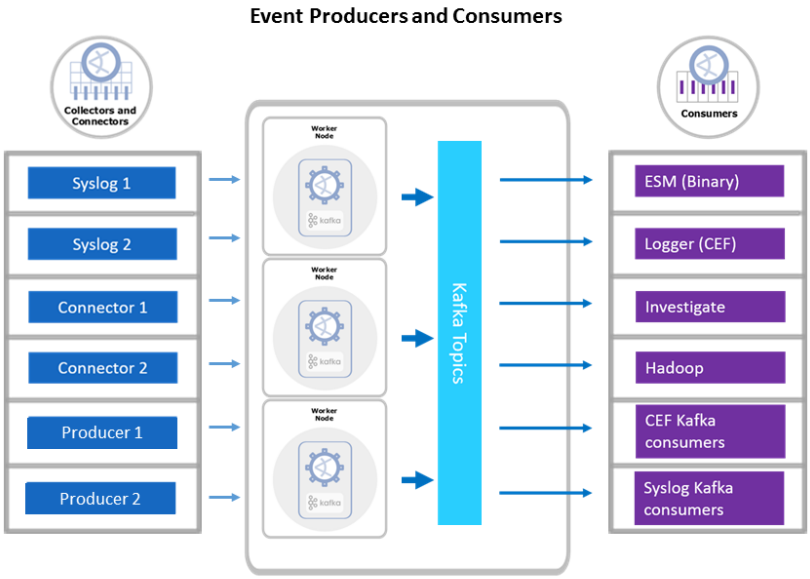
Transformation Hub ingests, enriches, normalizes, and then routes data from data producers to connections between existing data lakes, analytics platforms, and other security technologies and the multiple systems within a corporate security operations center. Transformation Hub can seamlessly broker data from any source and to any destination. Its architecture is based on Apache Kafka and it supports native Hadoop Distributed File System (HDFS) capabilities, enabling both the Logger and ArcSight Investigate technologies to push to HDFS for long-term, low-cost storage.

The latest releases of ArcSight Investigate are integrated with the Transformation Hub for raw events, as well as integrated with ESM to receive alerts and start the investigation process.

ESM receives binary events for dashboarding and further correlation.

This architecture reduces the overall infrastructure footprint, scales event ingestion using built-in capabilities and greatly simplifies upgrades to newer Transformation Hub releases. It also positions the platform to support an analytics streaming plug-in framework, supporting automated machine learning and artificial intelligence engines for data source onboarding, event enrichment, and entities and actors detection and attribution.

The diagram shows the broad scheme of event data flow through the Transformation Hub and associated products. The ***CDF Planning Guide*** contains a detailed discussion of deployment architectures.



Secure Open Data Platform (SODP)

The Secure Open Data Platform (SODP) centralizes management, monitoring and configuration of the entire data-centric ecosystem using an open architecture. It is configured and monitored through the ArcSight Management Center (ArcMC) user interface.

SODP comprises the following ArcSight products:

- Transformation Hub
- Management Center (ArcMC)
- Smart Connectors
- Logger

Management Center (ArcMC)

ArcMC is a central administrative user interface for managing SODP components. This management console administers SODP infrastructure, including users, configurations, backups, updates, and health monitoring to connectors and storage instances. ArcMC's Topology view shows administrators event flow through the entire environment, including a specific focus on monitoring endpoint device log delivery.

SmartConnectors

SmartConnectors serve to collect, parse, normalize and categorize log data. Connectors are available for forwarding events between and from Micro Focus applications like Transformation Hub and ESM, enabling the creation of multi-tier monitoring and logging architectures for large organizations and for Managed Service Providers.

The connector framework on which all SmartConnectors are built offers advanced features that ensures the reliability, completeness, and security of log collection, as well as optimization of network usage. Those features include: throttling, bandwidth management, caching, state persistence, filtering, encryption and event enrichment. The granular normalization of log data allows for the deterministic correlation that detects the latest threats including Advanced Persistent Threats and prepares data to be fed into machine learning models.

SmartConnector technology supports over 400 different device types, leveraging ArcSight's industry-standard Common Event Format (CEF) for both Micro Focus and certified device vendors. This partner ecosystem keeps growing not only with the number of supported devices but also with the level of native adoption of CEF from device vendors.

Logger

Logger provides proven cost-effective and highly-scalable log data management and retention capabilities for the SIEM, expandable to hundreds of nodes and supporting parallel searches. Notable features of Logger includes:

- Immutable storage
- High compression
- Archiving mechanism and management
- Transformation Hub integration
- Advanced reporting wizard
- Deployed as an appliance, software or cloud infrastructure
- Regulatory compliance packages

ArcSight Investigate

ArcSight Investigate simplifies security investigations using advanced analytics to proactively hunt for and defeat unknown threats to decrease the impact of security incidents. Powered by Micro Focus Vertica's high-performance database, and including integration with Hadoop, threat analysis is simplified using built-in Vertica-based analytics in a dashboard-driven and intuitive hunt interface.

Chapter 2: Producing and Consuming Event Data

Transformation Hub's publish-subscribe messaging system uses SmartConnectors and Collectors to produce event data, and supports Logger, ArcSight Investigate, and ESM, as well as Apache Hadoop and other third-party consumers.

While Transformation Hub can support a very high event flow (millions of events per second), the event rate for each producer and consumer will generally be much smaller (tens of thousands of events per second). Actual event flow will depend on your specific implementation and tuning applied, as well as server resources available (such as memory and CPU).

This section includes the following topics:

Producing Events with SmartConnectors

SmartConnectors can publish events to Transformation Hub topics. In order to publish events, you must configure your SmartConnectors to use the Transformation Hub destination. To send events to multiple topics, you can configure multiple concurrent destinations with the same Transformation Hub using different topics.

Once configured with a Transformation Hub destination, the SmartConnector sends events to Transformation Hub's Kafka cluster, which can then further distribute events to real-time analysis and data warehousing systems. Other applications, including ArcSight Investigate, ESM, Logger, and any third-party application that supports retrieving data from Kafka can receive them, for example, Apache Hadoop.

Transformation Hub balances incoming events between nodes, by distributing them evenly between the partitions in the configured topic.

Acknowledgments ("acks") ensure that Transformation Hub has received the event before the SmartConnector removes it from its local queue. You can disable acknowledgments, require acknowledgment only from the primary replica, or require every replica to acknowledge the event. (Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has.)

Note: Performance impact due to leader acks is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

For CEF topics, supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information, such as Logger Device Groups.

- For information on supported SmartConnector versions, ,see the ***SODP Support Matrix.***
- For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the ***SmartConnector User's Guide.***

Micro Focus documentation is is available for download from the the Micro Focus support community.

Consuming Events with ESM

ESM agents are the consumers for Transformation Hub's publish-subscribe messaging system. An ESM agent can connect to Transformation Hub and consume all events in binary format for the topics it subscribes to.

Additionally, ESM provides data monitors to monitor Transformation Hub health.

- For information on supported versions of ESM and SmartConnectors, see the ***SODP Support Matrix.***
- For instructions on configuring a supported version of ESM as a consumer, see the ***ArcSight ESM Administrator's Guide.***

Micro Focus documentation is available for download from the Micro Focus support community.

Consuming Events with Logger

To subscribe to Transformation Hub topics with Logger, you must configure a receiver on a supported Logger version to receive the Transformation Hub events. Logger's Transformation Hub receivers are consumers for Transformation Hub's publish-subscribe messaging system. They receive events in Common Event Format (CEF) from Transformation Hub topics. A Logger Transformation Hub receiver connects to Transformation Hub and consumes all events for the topics it subscribes to.

When configuring a Logger Transformation Hub receiver, specify the worker node FQDNs, topics to consume from, and consumer group name. You can configure multiple Loggers to consume from the same topic as a part of a consumer group.

For more information about Logger and how to configure a Transformation Hub receiver, refer to the ***Logger Administrator's Guide.,*** available for download from the Micro Focus support community.

Note: Kafka consumers can take up to 24 hours for the broker nodes to balance the partitions among the consumers. Check the Transformation Hub Kafka Manager **Consumers** page to confirm all consumers are consuming from the topic.

Sending Transformation Hub Data to Logger

For a Logger to be able to consume Transformation Hub events, the Logger must have a Transformation Hub receiver configured with the Transformation Hub worker nodes, consumer group, and event topic list. SmartConnectors that send data to Transformation Hub must have a Transformation Hub destination.

A group of Loggers, called a pool, can be configured to receive and distribute events between themselves. This works similarly to the Logger pool created by using the Logger Smart Message Pool destination on SmartConnectors. The difference is that when the SmartConnectors have a Logger Smart Message Pool destination, the event load is balanced by each SmartConnector, but when the SmartConnectors have a Transformation Hub destination, the event load is balanced by the Loggers.

Additional Loggers can be added to the pool simply by configuring the same Transformation Hub worker nodes, consumer group, and event topic list in the new Logger's Transformation Hub receivers, without having to reconfigure either the existing Loggers or any SmartConnectors.

The events retrieved by the Logger pool are distributed among the Loggers in the pool. If one Logger is down, new events are rebalanced among existing Loggers. When a Logger is added or removed from the Consumer Group, the event load is distributed across the pool of Loggers.

To send events from a group of SmartConnectors to a pool of Loggers, configure their Transformation Hub destinations to send events to the topic from which the Logger pool is consuming.

To configure Logger to subscribe to event data from specific SmartConnectors, you can do either of the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic.
- Configure each SmartConnector to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

Tip: Loggers in the same Logger pool do not consume the same events, since they are in the same Consumer Group. In high availability situations, you need events to be stored on two different Loggers. To store the same events on two Loggers, configure the Loggers to have different Consumer Group names, but subscribe them to the same event topic.

The number of Loggers in a Logger pool is restricted by the number of event topic partitions configured on the Transformation Hub. For example, if there are only five partitions configured, only five Loggers will receive the events. If you have more than five Loggers configured in the same Consumer Group, some Loggers will not normally receive events, but will be available as hot spares. When adding receivers, be sure to increase the number of event topic partitions. See Managing Topics for more information.

Sending Transformation Hub data to Logger (Overview):

1. Configure the SmartConnector:
   - Set up a SmartConnector to publish to a particular Transformation Hub topic. Connectors can only send to a single topic for each destination. Additional destinations need to be configured if each event needs to go to multiple topics. Note the number of partitions in the topic.
   - For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the Micro Focus support community.
2. Configure Logger:
   - Create a Transformation Hub receiver on each Logger in the Logger pool.
   - Configure each receiver to subscribe to the topics to which the SmartConnectors are publishing data. To subscribe to multiple topics, indicate the topics by specifying them in the Event Topic List parameter (a list of comma-separated values) while configuring the Transformation Hub receiver.
   - Configure each receiver to be in the same Consumer Group.

Example Setup with Multiple Loggers in a Pool

You can set up your Logger pools to subscribe to events from a particular device type, such as "Firewall." To do this, you would:

1. In ArcMC, create a Kafka topic named ***Firewall***.
2. Configure all the SmartConnectors that handle firewall events to publish these events to topic "Firewall."
3. Configure the Loggers in the Logger pool:
   - Create a Transformation Hub Receiver on each Logger in the pool.
   - Configure the receivers to subscribe to the event topic “Firewall,” and include them in the “Logger_Firewall” Consumer Group.

Once the configuration is set up properly, the Logger pool will subscribe to device type ***Firewall***.

Note: This example assumes that the Transformation Hub is being managed by an ArcSight Management Center for topic creation. Topics can also be managed through the Kafka Manager UI.

Consuming Events with ArcSight Investigate and Micro Focus Vertica

Transformed events in the default topic **`th-arcsight-avro`**, which are in Avro format, can be read by the Micro Focus Vertica database. In turn, once in Vertica storage, event data is accessible for use in ArcSight Investigate searches.

You configure ArcSight Investigate for use with Transformation Hub as part of the Vertica installer, where you can specify the location of the default Avro topic to which Vertica can subscribe. Consult the ***ArcSight Investigate Deployment Guide***, available from the Micro Focus support community, for how to set up the Vertica scheduler for the Investigate use case.

Consuming Events with Third-Party Applications

Transformation Hub is designed with support for third-party tools. You can create a standard Kafka consumer and configure it to subscribe to Transformation Hub topics. By doing this you can pull Transformation Hub events into your own data lake.

Note: Custom consumers must use Kafka client libraries of version 0.11 or later.

- All Transformation Hub nodes, consumers, and producers must be properly configured for forward /reverse DNS lookup, as well as time-synchronized, using a time server such as NTP.
- Events are sent in standard CEF (CEF text) and binary (exclusively for ESM consumption). Any software application that can consume from Kafka and understand CEF text can process events.
- You can set up multiple consumer groups, and each group will get a copy of every event. Therefore you can have Logger and Apache Hadoop configured to consume from the same topic and each will get a copy of every event. This enables fanning out multiple copies of events without reconfiguring SmartConnectors or using additional CPU or network resources for them.

Consuming Transformation Hub Events with Apache Hadoop

Apache Hadoop is a software framework that enables the distributed processing of large data sets across clusters of computers. You can send Transformation Hub events to Hadoop by using Apache Flume.

This section describes how to set up the Apache Flume agent to transfer Common Event Format (CEF) events from an Transformation Hub Kafka cluster to Hadoop Distributed File System (HDFS).

It includes the following topics:

Architecture for Kafka to Hadoop Data Transfer

Apache Flume uses a source module to read a Kafka topic containing CEF events, and it then transfers the events using a memory channel, and persists them to HDFS using a sink module. The CEF files are stored on HDFS by time, in a year/month/day/hour directory structure.

Using Apache Flume to Transfer Events to Hadoop

One of the applications you could use to transfer Transformation Hub events into your data lake is Apache Flume. Flume is designed to push data from many sources to the various storage systems in the Hadoop ecosystem, such as HDFS and HBase. This section describes how to use Apache Flume as a data transfer channel to transfer events from Transformation Hub to Apache Hadoop or other storage systems.

Prerequisites

- Transformation Hub installed: Consult the ***Micro Focus Transformation Hub Deployment Guide***.
- Flume installed: For information on how to install and configure Flume, refer to the Flume documentation, available at https://flume.apache.org/releases/content/1.6.0/FlumeUserGuide.pdf.
- Storage system installed: Refer to your storage system documentation.

Procedure

Flume is controlled by an agent configuration file. You must configure Transformation Hub as the source agent, your storage system as the sink agent, and ZooKeeper as the channel agent in this file.

To configure Transformation Hub as the source:

Edit the agent configuration file to include the required properties, as in the table below. Configure other properties as needed for your environment.

Required Kafka Source Configuration

Property	Description
type	Set to `org.apache.flume.source.kafka.KafkaSource`.
topic	The Event Topic from which this source reads messages. Flume supports only one topic per source.

To configure the sink:

The required configuration varies. Refer to the Flume documentation for details on your storage system. The section Consuming Events with Apache Flume provides an example of how to configure Apache Hadoop as the sink.

Setting Up Flume to Connect with Hadoop

In the simplest deployment model, you need to deploy the Apache Flume agent on a Hadoop node server to pull events, and send them to Hadoop Distributed File System (HDFS).

Hadoop must be installed before you can connect it with Flume. If you do not already have your own Hadoop deployment, you can deploy Hadoop on a Red Hat Enterprise Linux 7.2 host. For more information, see Setting Up Hadoop.

To set up Flume to connect with Hadoop:

1. Log into your Hadoop server as the user "hadoop".
2. Download Flume from the Apache download site.
3. Uncompress the ".gz" file to your preferred deployment directory.
4. In the configuration file, add your Kafka bootstrap server addresses and port numbers, Kafka topic, and HDFS address and port..

   By default, this configuration persists a CEF file every hour. If you have a high volume of events, use the event count option instead of time, to avoid running out of memory. For more information, refer to documentation of the Flume HDFS sink in the Flume Users' Guide.
5. Execute the following commands to create the Hadoop cefEvents directory:
   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```
6. Create a configuration file in the Flume conf directory, `bin/flume/conf/`, following the template in Sample Flume Configuration File. In our example, the file is named ***kafka.conf***. You can name it whatever is appropriate.
   a. Copy `flume-env.sh.template` as `flume-env.sh`.
   b. Edit `flume-env.sh` file and make the following changes:
      - Set JAVA_HOME to the directory where Java was installed on your system.
      - Uncomment the line for JAVA_OPTS:

        ```
        export JAVA_OPTS="-Xms100m -Xmx2000m -Dcom.sun.management.jmxremote"
        ```
      - Set `FLUME_CLASSPATH=<Flume install directory>/lib`.
   c. Copy the common jar files from the Hadoop install directory to the Flume lib directory:

      ```
      cp <Hadoop install directory>/share/hadoop/common/*.jar /<Flume Install directory>/lib
      ```

      ```
      cp <Hadoop install directory>/share/hadoop/common/lib/*.jar /<Flume Install directory>/lib
      ```
   d. Copy hadoop-hdfs-2.7.2.jar from the Hadoop install directory to the Flume lib directory.

```
cp <Hadoop install directory>/share/hadoop/hdfs/hadoop-hdfs-2.7.2.jar
/<Flume Install directory>/lib
```

7. Execute the following command to start Flume from its home directory:
   `bin/flume-ng agent --conf conf/ --conf-file conf/kafka.conf --name tier1 -Dflume.root.logger=INFO,console`
8. After you start Flume, you can find the files on HDFS by running the following command:
   `hadoop fs -ls -R /opt/hadoop/cefEvents`

   This path has to match the HDFS directory path, created in the Hadoop configuration section.

   The files are stored in following structure: "year/month/day/hour".

Sample Flume Configuration File

Before starting Apache Flume, create a configuration file based on the template below.

The configuration file should reside in `bin/flume/conf/`. This file is called *kafka.conf* in our example. You can name your own configuration file whatever is appropriate.

```
#####################################################

#Sample Flume/Kafka configuration file

#####################################################

#defines Kafka Source, Channel, and Destination aliases

tier1.sources = source1

tier1.channels = channel1

tier1.sinks = sink1

#Kafka source configuration

tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource

tier1.sources.source1.kafka.bootstrap.servers= kafkaIP1:9092, kafkaIP2:9092,…

tier1.sources.source1.kafka.topics = th-cef

tier1.sources.source1.kafka.consumer.group.id = flume

tier1.sources.source1.channels = channel1

tier1.sources.source1.interceptors = i1

tier1.sources.source1.interceptors.i1.type = timestamp

tier1.sources.source1.kafka.consumer.timeout.ms = 150

tier1.sources.source1.kafka.consumer.batchsize = 100
```

```
#Kafka Channel configuration

tier1.channels.channel1.type = memory

tier1.channels.channel1.capacity = 10000

tier1.channels.channel1.transactionCapacity = 1000

#Kafka Sink (destination) configuration

tier1.sinks.sink1.type = hdfs

tier1.sinks.sink1.channel = channel1

tier1.sinks.sink1.hdfs.path = hdfs://localhost:9000/opt/\

hadoop/cefEvents/year=%y/month=%m/day=%d

tier1.sinks.sink1.hdfs.rollInterval = 360

tier1.sinks.sink1.hdfs.rollSize = 0

tier1.sinks.sink1.hdfs.rollCount = 0

tier1.sinks.sink1.hdfs.fileType = DataStream

tier1.sinks.sink1.hdfs.filePrefix = cefEvents

tier1.sinks.sink1.hdfs.fileSuffix = .cef

tier1.sinks.sink1.hdfs.batchSize = 100

tier1.sinks.sink1.hdfs.timeZone = UTC
```

Setting Up Hadoop

This is an overview of the steps necessary to install Apache Hadoop 2.7.2 and set up a one-node cluster. For more information, see https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/SingleCluster.html, or refer to the Hadoop documentation for your version.

To install Hadoop:

1. Be sure that your environment meets the operating system and Java prerequisites for Hadoop.
2. Add a user named 'hadoop'.
3. Download and unpack Hadoop.
4. Configure Hadoop for pseudo-distributed operation.
   - Set the environment variables.
   - Set up passphraseless SSH.

- Optionally, set up Yarn. (You will not need Yarn if you want to use Hadoop only storage and not for processing.)
- Edit the Hadoop configuration files to set up a core location, a Hadoop Distributed File System (HDFS) location, a replication value, a NameNode and a DataNode.
- Format the Name node.

5. Start the Hadoop server using the tools provided.
6. Access Hadoop Services in a browser and login as the user "hadoop".
7. Execute the following commands to create the Hadoop cefEvents directory:
   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```
8. Execute the following commands to grant permissions for Apache Flume to write to this HDFS
   ```
   hadoop fs -chmod 777 -R /opt/hadoop
   hadoop fs -ls
   ```
9. Execute the following command to check Hadoop system status:
   ```
   hadoop dfsadmin -report
   ```
10. Execute the following command to view the files transferred by Flume to Hadoop.
    ```
    hadoop fs -ls -R /
    ```

Connectors in Transformation Hub

Transformation Hub includes support for Connectors in Transformation Hub (CTH). CTH moves the security event normalization, categorization, and enrichment of connectors processing to the Docker containers environment of Transformation Hub, and leaves the remaining workload, raw data collection, to a component called a Collector. Up to 50 CTHs may be deployed in a Transformation Hub cluster. Only syslog data is supported.

Deploying CTHs is performed using ArcMC 2.92 or later, managing a Transformation Hub. For information on managing Transformation Hub and deploying CTH, see the ***Micro Focus ArcSight Management Center Administrator's Guide,*** available from the Micro Focus support community.

Configuring Consumers and Producers for Availability

Configure the Transformation Hub Kafka cluster endpoint to avoid single points of failure in both the producers sending data to Transformation Hub (such as connectors), and the consumers subscribing to data from the Transformation Hub (such as Logger).

For Producers

Configure the **Initial Host:Port(s)** parameter field in the Transformation Hub Destination to include all Kafka broker (worker) nodes as a comma-separated list.

Provide all Kafka broker (worker)nodes for a producer and a consumer configuration to avoid a single point of failure.

For more information on how Kafka handles this using bootstrap.servers, please see:

https://kafka.apache.org/documentation/#producerconfigs.

For Consumers

Configure the **Transformation Hub host(s) and port** parameter field in the Receiver to include all Kafka cluster nodes as a comma-separated list.

For more information on how Kafka handles this using bootstrap servers, please see:

https://kafka.apache.org/documentation/#consumerconfigs.

Chapter 3: Securing your Transformation Hub deployment

You are responsible for configuring your Transformation Hub environment securely according to your business needs and requirements. To help you do this, the Transformation Hub supports Transport Layer Security (TLS) 1.2.

This section includes the following topics:

Changing Transformation Hub Security Mode

You should decide on a security mode for Transformation Hub prior to deployment and setup. In general, the security mode of systems connected to Transformation Hub (consumers and producers) must be the same as the Transformation Hub security mode.

TLS is the default security mode. Optional modes include TLS with Client Authentication, as well as FIPS. TLS performance impact is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

You can change the Transformation Hub security mode after deployment, but this will cause downtime for your Transformation Hub and associated systems, such as consumers and producers. You will need to make sure all Transformation Hub-associated systems are re-configured as well. If the security mode change requires that Transformation Hub consumer or Transformation Hub producer restarts, the ***producer or consumer must be disconnected from Transformation Hub first.*** Consult the appropriate consumer or producer documentation for details.

The process of changing security mode includes the following steps.

Note: ***Undeploying Transformation Hub will remove all previous configuration settings.*** Prior to proceeding further, you should make a note of your existing settings and then re-enter these on the pre-deployment configuration page during the re-deployment of the Transformation Hub.

1. Stop SmartConnectors from sending events. This will close connections. See the SmartConnector User Guide for information on stopping SmartConnectors from sending events.
2. Stop all consumers (ArcSight Logger, ArcSight ESM, Vertica Scheduler) from consuming from topics in Transformation Hub. (There is no need to clear out existing messages from the topics, and the consumers will pick up where they left off later.)
3. Log in to the CDF Management Portal (`https://<ha-address>:5443`).
4. Click **Administration**.

5. Click the **...** (Browse) icon to the right of the main window.
6. From the drop-down, click **Uninstall**. The post-deployment settings page is displayed.
7. Uninstall the Transformation Hub.
8. Follow the consumer and producer documentation to reconfigure those applications to align their security modes to be the same as Transformation Hub.
9. Redeploy the Transformation Hub with the appropriate security mode configured, as outlined in the *Transformation Hub Deployment Guide*.
10. Reconnect the consumers and producers to the Transformation Hub. See the respective product documentation for these procedures.

Chapter 4: Managing Transformation Hub

You can manage topic routing and Transformation Hub infrastructure through ArcMC. Additionally, Transformation Hub provides the open-source Transformation Hub Kafka Manager to help you monitor and manage its Kafka services.

- For more information about Kafka Manager, refer to https://github.com/yahoo/CMAK.
- For more information about Kafka monitoring, refer to the monitoring section of the Kafka documentation.

This section includes the following topics:

Licensing Transformation Hub

Transformation Hub ships with a 90-day trial license, which will enable functionality for the trial period. To ensure continuous event flow and uninterrupted service, you should ensure that you apply the correct license to Transformation Hub as soon as possible after initial deployment.

Transformation Hub supports Transformation Hub licenses, as well as legacy ADP ArcMC licenses (formerly valid for the legacy Event Broker application).

You should make sure you have obtained your valid license file from Micro Focus before performing this procedure.

To install your license:

1. Log in to the Management Portal (`https://<ha-address>:5443`).
2. Click **Suite**.
3. Click the **...** (Browse) icon to the right of the main window.
4. From the drop-down, click **License**. The **License Management** page is displayed.
5. Optionally, select the **I authorize Micro Focus to collect suite and product data...** checkbox to send usage data to Micro Focus to help improve the product.
6. Under **Install Licenses**, click **Choose File.**
7. Browse to the location of your valid license file., and then click **Next**.
8. Follow the prompts to apply your license.
9. After applying your license file, restart each Kafka pod in the cluster, ***one at a time,*** as follows:

- For each of the Kafka pod from 0 to x, restart the selected Kafka pod with the command:
  `kubectl delete pod th-kafka-(x) -n arcsight-installer-XXX`
- Watch the logs and ensure the Kafka pod is up and running by running this command:
  `kubectl logs th-kafka-(x) -n arcsight-installer-XXX`

Note: You can also check the status of the restarted broker node using the Transformation Hub Kafka Manager.

Once the selected broker node is up and running, only then proceed to restart the next node.

License Verification

For each Kafka broker node, the license check result is logged both in the Kafka pod log and in the file `/opt/arcsight/k8s-hostpath-volume/th/autopass/license.log`. If there is a valid license, the log will include the text:

```
TH licensed capacity: <eps number>
```

If no license has been installed, this text will be included instead:

```
ERROR: No valid license key was found. Please install a valid license key or
contact Micro Focus Customer Support for instructions on how to get one.
```

Managing Transformation Hub through ArcMC

After configuring ArcMC to manage your Transformation Hub, you can create topics and routing rules, monitor Transformation Hub metrics, and receive notifications about Transformation Hub status through ArcSight Management Center (ArcMC).

Monitored Transformation Hub parameters include CPU usage, memory, event parsing errors, stream processing EPS, and stream processing lag.

To manage a Transformation Hub in ArcMC, add your Transformation Hub as a host to ArcMC. The procedure for adding Transformation Hub as a host is explained in detail in the ***Micro Focus ArcSight Management Center Deployment Guide***, available from the Micro Focus support community.

Note: A single ArcMC can manage a single Transformation Hub cluster.

The ***Micro Focus ArcSight Management Center Administrator's Guide*** also explains in detail how to view the status of Transformation Hub consumers, as well as how to manage topics, routing rules, and monitored metrics.

About the Transformation Hub Kafka Manager

The Transformation Hub Kafka Manager enables you to monitor and manage your clusters, topics, and partitions, including the following:

- Viewing and managing cluster states, including topics, consumers, offsets, broker nodes, replica distribution, and partition distribution.
- Creating and updating topics.
- Generating partitions and adding partitions to a topic.
- Reassigning partitions to other broker nodes, such as replacing a failed node with a new one.
- Reassigning partition leaders to their preferred broker node after a node temporarily leaves the cluster (for example, in case of a reboot).
- Managing JMX polling for broker-level and topic-level metrics.

Connecting to the Transformation Hub Kafka Manager

Only users that can log into the Transformation Hub server can access the Transformation Hub Kafka Manager. These users can access the Transformation Hub Kafka Manager by using their local web browser directly from any of the Transformation Hub nodes or by using SSH forwarding from a local system.

You can connect to the Transformation Hub Kafka Manager with the Chrome or Firefox browsers. For a list of browser versions supported in this release, refer to the ***CDF Planning Guide,*** available for download from the Micro Focus Support Community.

To access Transformation Hub Kafka Manager:

1. On a Transformation Hub node, run the command to get the Kakfka Manager service.:
   `kubectl get services --all-namespaces|grep th-kafkamgr-svc`
2. Note `th-kafkamgr-svc` service and note its IP and port number.

To connect directly from a Transformation Hub node:

1. Log into the Transformation Hub node.
2. In a terminal window, run the following command:
   `kubectl -n <the arcsight-installer-* namespace> port-forward <the th-kafka-manager-* pod name> 9000:9000`
3. With a supported browser, connect to Transformation Hub Kafka Manager:
   `http://localhost:9000`

Once connected, the browser displays the **Clusters** page. For more information on this page, see Managing Clusters.

To connect from your local host:

1. From your local system, set up SSH forwarding and connect by using a command like the following:

   ```
   ssh -L <local port>:<Transformation Hub Kafka Manager Service IP>:<port> root@<TH master node address>
   ```

2. With a supported browser, connect by using the following URL:

   ```
   http://<Transformation Hub Kafka Manager Service IP>:<local port>
   ```

Once connected, the browser displays the **Clusters** page. For more information on this page, see Managing Clusters.

Managing Clusters

The **Clusters** page is the Transformation Hub Manager's home page. From here you can modify, disable or delete a cluster from view in the Transformation Hub Manager (the cluster itself is not deleted), or drill down into the cluster for more information.

Location: Clusters

Click the *Cluster Name* link. The Transformation Hub Manager displays the **Cluster Summary** page. For more information, see Viewing Information About a Cluster.

To edit the cluster:

1. Click **Modify**. The Transformation Hub Manager displays the **Update Cluster** page.
2. Update the appropriate fields, and click **Save**.

Editing the cluster is an advanced operation, and normally the cluster should never be edited.

To disable the cluster:

Click **Disable**. Once a cluster has been disabled, a **Delete** button is displayed.

To delete the cluster:

After disabling the cluster, click **Delete**.

Viewing Information About a Cluster

On the **Summary** page, you can view the ZooKeeper processes in your cluster and drill down into its topics and broker nodes for more information.

Location: Clusters > *Cluster Name* > Summary

To view information about your cluster:

- If the cluster is not yet open, click **Cluster > List** in the navigation bar. Then click the *Cluster Name* link.
- If the cluster is already open, click **Clusters > *Cluster Name* > Summary**

To view or edit the topics in your cluster:

Click the **Topics** hyperlink (number of topics) to show the topics in the cluster. For more information, see Managing Topics.

To view or edit the broker nodes in your cluster:

Click the **Brokers** hyperlink (number of broker nodes) to show the broker nodes in the cluster. For more information, see Managing Brokers.

Managing Brokers

On the **Brokers** page, you can see an overview of all of your Worker nodes and drill down into a node for more information.

> **Note:** The term *Brokers* refers to nodes running Kafka services (that is, Kubernetes worker nodes, but not master nodes).

Location: Clusters > *Cluster Name* > Brokers

To view the broker nodes in your cluster:

Click **Brokers** in the navigation bar. The **Brokers** page opens.

To see more information about a specific broker:

Click the broker's ***Id*** link. The ***Broker Name*** ID opens. For more information, see Viewing Broker Details.

Viewing Broker Details

You can view detailed information about a broker from the ***Broker Name*** details page.

Location: Clusters > *Cluster Name* > Brokers > *Broker Name*

To view information on a specific broker:

1. Click **Brokers** in the navigation bar.
2. Click the *Broker Name* link. The *Topic Name* page opens.

The following data is displayed.

Summary

In the **Summary** section, you can see an overview of your broker, including the number of topics and partitions located on it.

Metrics

In the **Metrics** section, you can view information about the data flow.

Messages count

In the **Messages** section, you can view a message view chart.

Per Topic Detail

In the **Per Topic Detail** section, you can view topic replication and partition information and drill down to view more information on each topic.

To see more information about a specific topic:

Click the *Topic Name* link in the **Per Topic Details** section. See Viewing Topic Details

Managing Topics

On the **Topics** page, you can run or generate partition assignments, add a new partition, and drill down into individual topics for more information.

Location: Clusters > *Cluster Name* Topic > List

Note: The following topics are created by default. They are used internally by Transformation Hub and should not be deleted, modified, or used by external data producers or consumers.

```
__consumer_offsets
 _schemas
th-arcsight-json-datastore
th-arcsight-avro-sp_metrics
th-syslog
th-arcsight-avro
```

To manage the topics in your cluster:

Click **Topic > List** in the navigation bar.

To view information on a topic:

Click the *Topic Name* link. The *Topic Name* page displays the topic's summary, metrics, consumers, and partitions. See Viewing Topic Details.

To generate partition assignments:

1. Click **Generate Partition Assignments**.
2. Select the topics and broker nodes to reassign.
3. Click **Generate Partition Assignments**.

To assign partitions as generated:

1. Click **Run Partition Assignments**.
2. Select the topics to reassign.
3. Click **Run Partition Assignments**.

To add a partition:

1. From the Topics Summary page, click **Add Partition**.
2. Enter the new number of partitions.
3. Select the topics and broker nodes.
4. Click **Add Partitions**.

Creating Topics

You can create a new topic on the **Create Topic** page.

Location: Clusters > *Cluster Name* Topics > Create Topic

To open the Add Topic page:

Click **Topic > Create** in the navigation bar.

To create a new topic:

1. Fill in values for the **Topic Name,** number of **Partitions**, and **Replication Factor** fields
2. Click **Create**.

For a discussion of field values, consult the Kafka documentation at https://kafka.apache.org/documentation/#topicconfigs.

The number of custom topics you can create will be limited by Kafka, as well as performance and system resources needed to support the number of topics created.

Note: Alternatively, you can also create topics on a managed Transformation Hub in ArcMC, or invoke the `kafka-topics` command from the CLI on the Kafka pod using the `kubectl exec` command.

Viewing Topic Details

You can see details about a topic, including information about the summary, metrics, consumers, and partitions from the *Topic Name* details page.

Location: Clusters > *Cluster Name* Topics > *Topic Name*

To view information on a specific topic:

1. Click **Topic > List** in the navigation bar.
2. Click the *Topic Name* link. The *Topic Name* page opens.

The following data is displayed.

Topic Summary

In the **Topic Summary** section, you view information on the topic's replicas, partitions, and broker nodes.

Metrics

In the **Metrics** section, you can view information about the data flow.

Operations

In the **Operations** section, you can perform a variety of tasks on dbroker nodes.

To reassign partitions:

Click **Reassign Partitions**.

To update a topic's configuration:

1. Click **Update Config**.
2. Edit the configuration fields.
3. Click **Update Config**.

To specify partition assignments:

1. Click **Manual Partition Assignment**.
2. Select the desired assignments.
3. Click **Save Partition Assignment**.

Partitions by Broker

In the **Partitions by Broker** section, you can see topic partition information and drill down to see details for each broker.

To view details on a broker:

Click the **Broker** link. The **Topic Summary** page displays information on the topic's lag, partitions, and consumer offset.

In Transformation Hub Kafka Manager, users will see different offset values between CEF (Investigate or Logger) topics and binary (ESM) topics. In CEF topics, the offset value can generally be associated with number of events that passed through the topic. Each message in a CEF topic is an individual event. However, that same association cannot be made for the ESM topic, as several events are batched into each messagen.

Consumers consuming from this topic

In the **Consumers consuming from this topic** section, you can drill down to see details on each consumer.

New consumers can take some time to display properly. Give the process time to populate correct data.

To view details on a consumer:

Click the *Topic Name* link. The Topic Summary page displays information on the topic's lag, partitions, and consumer offset.

Partition Information

In the **Partition Information** section, you can view information about the topic's partitions and drill down for more information on each leader.

To view details on a leader:

Click the **Leader** link. The *Broker Name* ID page displays the broker's summary, metrics, message count, and topic details. See Viewing Broker Details.

Managing Consumers

On the **Consumers** page, you can see a list of consumers, view their type, the topics they consume, and drill down into each consumer and topic for more information.

Location: Clusters > *Cluster Name* > Consumers

To view or edit the consumers in your cluster:

Click **Consumers** in the navigation bar.

To view more details on a specific consumer:

Click the *Consumer Name* link. The *Consumer Name* page displays details about the consumer. You can drill down further for more information, including Consumed Topic Information (such as Partitions Covered % and Total Lag).

To view more details on the topic it consumes:

Click the *Topic Name* link. The *Topic Name* page displays details about the topic. You can drill down further for more information including Consumer Lag, and Consumer Offset and LogSize data by Partition.

Viewing Consumer Details

You can see a information about a consumer and drill down on the topics it consumes from the ***Consumer Name*** details page.

Location: Clusters > *Cluster Name* Consumer > *Consumer Name*

To view information on a consumer:

1. Click Clusters > *Cluster Name* Consumer.
2. Click the *Consumer Name*.

To view information on the consumed topic:

1. Click the *Topic Name*. The Consumed Topic Information page displays information about the topic. Click the topic name for more information including Consumer Lag and Consumer Offset and LogSize data by partition .

Managing Preferred Replicas

You can update the replicas for each cluster on the **Preferred Replica Election** page.

Location: Clusters > *Cluster Name* > Preferred Replica Election

To open the Preferred Replica Election page:

Click **Preferred Replica Election** in the navigation bar.

To run the Preferred Replica Election for your topic:

Click **Run Preferred Replica Election**.

Managing Partitions

You can reassign partitions for your cluster on the **Reassign Partitions** page.

Location: Clusters > *Cluster Name* > Reassign Partitions

To open the Reassign Partitions page:

Click **Reassign Partitions** in the navigation bar.

To reassign the partitions for your topic:

Click **Reassign Partitions**.

Configuring Topic Partitions Based on Number of Consumers

You can scale the consumption rate for a consumer of a topic by adding more consumers to the consumer group. However, when adding new consumers to the consumer group, please consider the topic partition count of the topic you are consuming from. The following table shows the relationship between the number of consumers in a consumer group and data consumption from each partition.

Number of Consumers in Group is...	Consumption from Partitions
A single consumer	Consumes from all partitions in source topic.
Lower than partition count	Each consumer consumes from a subset of the topic partitions.
Equals partition count	Each consumer consumes from each of the topic partitions.
Exceeds partition count	Each consumer consumes from each of the topic partitions; additional consumers stay idle until new partitions are added to the source topic.

If you change the number of partitions in the source topic to match the consumer group size (same or a multiple) for a given consumer group consumption rate, or add additional consumers in the consumer to match the topic partition count, then the Transformation Hub will automatically re-balance the consumer groups.

Graceful Shutdown and Rebooting of Transformation Hub Nodes

Shutting down and rebooting of Transformation Hub nodes is required by corporate IT for tasks such as OS patches and updates. If the procedure described below is not used and the nodes of your Transformation Hub are forcefully shut down and restarted, system corruption may result.

Note: No data is lost when nodes are shut down gracefully.

To gracefully reboot each cluster node (master nodes first, worker nodes last), and avoid disrupting the flow of events, run the following commands:

```
ssh <node_ip/hostname>

kube-stop.sh
```

Note: A known issue exists where the `kube-stop.sh` script can misinform the user that services have been stopped. However, in reality, the services have not been stopped. **Monitor your processes to ensure the Docker and cluster processes are stopped before rebooting.** Restarting without

ensuring the services are stopped may impact the cluster's health. After you are sure the services are stopped, you may continue with the following commands.

```
sync; sync

systemctl is-active docker kubelet kube-proxy

reboot

kube start.sh
```

To monitor all pods and ensure they are back in running state before moving on to the next node, run:
`watch kubectl get pods --all-namespaces -o wide`

Adding a New Worker Node

You can add a new worker node to an existing Transformation Hub installation.

Note: The procedure described here will require re-deployment of Transformation Hub

To add 1 or more new worker nodes:

1. Set up and provision the new node according to the guidelines and system requirements given in the *Micro Focus CDF Planning Guide*. Note the IP address of the new node for use in the following procedures.
2. Modify your NFS server settings to add the new node to the `/etc/exports` file.

Note: Refer to the NFS server section of the CDF Planning Guide for more information.

3. Run the following to update the shared volumes:
   `exportfs -ra`
4. Log in to the CDF Management Portal (`https://<ha-address>:5443`).
5. Click **Administration**.
6. Click the **...** (Browse) icon to the right of the main window.
7. From the drop-down, click **Reconfigure**. The post-deployment settings page is displayed. Note any settings you have changed from default values.
8. Click **Uninstall** and confirm the uninstallation of the Transformation Hub. The capability is uninstalled.

Note: Existing topics and route data will not be lost.

9. On the **Add Node** screen, click the **+**(Add) icon and configure the additional nodes.

10. On the pre-deployment configuration page, increment each of the following properties by the number of added worker nodes:
    - # of Worker Nodes in the Kafka cluster
    - # of Worker Nodes running ZooKeeper in the Kafka cluster
11. Click **Save** to redeploy the Transformation Hub with the new settings.

Now log into the CDF Management Portal and add the appropriate labels to the new worker nodes. Refer to the Transformation Hub Deployment Guide for the details on labeling nodes

Next, to assign partitions to the new node:

1. Launch and connect to Transformation Hub Kafka Manager.

Note: Refer to Connecting to the Transformation Hub Kafka Manager for more information.

1. In **Cluster > Transformation Hub > Topics,** click **Generate Partition Assignments.**
2. On the **Confirm Assignments** page, confirm partition assignments for the new node and click **Generate Partition Assignments.**
3. On the main toolbar, click **Topic > List.**
4. Click **Run Partition Assignments.**
5. On the **Run Assignments** page, confirm partition assignments and click **Run Partition Assignments.**
6. The partition reassignment process begins. On the Reassign Partitions page, under **Status**, check for a date and time of the job completion to verify completion of the task.

The new worker nodes will now begin processing events data.

Backing Up and Restoring Master Nodes

Backup

Select and implement a third-party backup tool to back up your cluster data. Always store backup data on a secure system separate from your cluster nodes

To perform a backup, on each master node, perform all of the operations below to backup the Transformation Hub cluster control plane as well as the data stored on the internal NFS server (if applicable):

1. Backup the cluster configuration. As sudo or root user, run:

```
  kubectl get cm base-configmap -n core -o yaml > {$k8s-home}/base-
configmap.$(date "+%Y%m%d-%H%M%S")
```

2. Backup the following directories and files:

```
/etc/exports

/etc/profile

/etc/profile.d/proxy.sh

/etc/security/limits.d/20-nproc.conf

/opt/arcsight/volumes/*    (NOTE: applicable only if NFS server is running
internally on the master)

/root/.bashrc

/root/.kube

/usr/bin/docker*

/usr/bin/kube*

/usr/bin/vault

/usr/lib/systemd/system/kubelet.service

/usr/lib/systemd/system/kube-proxy.service

/usr/lib/systemd/system/docker.service

/usr/lib/systemd/system/docker.service.d/http_proxy.conf

/usr/lib/systemd/system/docker-bootstrap.service

/usr/lib/systemd/system/docker-bootstrap.service.d/http_proxy.conf
```

All directories and files under folder `$k8s-home` except the `$k8s-home/data` directory.

3. Backup the `etc` datastore.

Note: This requires temporarily stopping `etcd`, and then restarting it after the backup.

4. As sudo or root user, stop `etcd`.:

```
docker -H unix:///var/run/docker-bootstrap.sock stop etcd_container
```

5. Backup all directories and files under `$k8s-home/data/etcd`
6. As sudo or root user, start `etcd`:

```
docker -H unix:///var/run/docker-bootstrap.sock start etcd_container
```

Restore

If some files were accidentally deleted, restore them from the most recent backup.

For example: If the file **$k8s-home/scripts/uploadimages.sh** was accidentally deleted, restore it from the backup.

Note: The restored files must have the same owner and permissions as the original files.

In the event of the loss of the master node in a single master deployment, please contact Micro Focus Support for help with the recovery.

Uninstalling a Master or Worker Node

To uninstall an existing master or worker node from the cluster, open an SSH connection to the node and run the following commands.

```
cd $k8s-home
./uninstall.sh
```

Then, reboot the node to complete node removal .

When removing the node from the cluster, make sure that the cluster will still have enough resources to host the product workload without the node you are removing . Also, make sure to keep sufficient nodes labeled by the product labels.

Effects on the Cluster

If a worker node is uninstalled, all events data will be stored on the node by default under **/opt/arcsight/k8s-hostpath-volume/th/kafka.**

If a master node is stopped or uninstalled, that node will be reported as unavailable to the cluster. All other functionality, including events processing on the worker nodes, will continue.

Note: From a multi-master cluster with 3 master nodes, you can safely remove only one master node. By removing one of three master nodes you will lose high availability, but the cluster will continue to function. If you remove two of three master nodes, the cluster may become unavailable, and you will then need to set up the cluster from scratch.

Removing a Crashed Worker Node

In case of a worker node failure, do the following:

1. Add a new worker node to replace the failed node before removing the crashed one.
2. Run the following command on one of the healthy nodes to delete the crashed node's IP address from the cluster:

   ```
   kubectl delete node <crashed_node_ip_FQDN>
   ```

Note: Such an action needs to be performed manually by the cluster administrator, because there is no way for the cluster to distinguish permanent node failure from temporary network connectivity outage, restart or similar events.

When this command is run, the cluster re-schedules the stateful containers (Kafka, ZooKeeper, routing stream processors) to the remaining machines matching the container requirements (labels, resources).

Replacing a Crashed Master Node

If one of the master nodes crashes in an HA cluster deployed in multi-master configuration, the cluster as well as associated products will remain functional. However, the HA functionality will be lost. In order to restore high availability, you need to restore the failed master node and re-add it to the cluster.

In addition to removing the crashed node (as described under Removing a Crashed Node), delete the crashed master node from **etcd** and restore it as follows:

1. Check the cluster health status by running:

```
$K8S_HOME/bin/etcdctl --cacert=K8S_HOME/ssl/ca.crt --cert=$K8S_
HOME/ssl/server.crt --key=$K8S_HOME/ssl/server.key --endpoints=
[https://MASTER1_IPV4_ADDRESS:4001,https://MASTER2_IPV4_
ADDRESS:4001,https://MASTER3_IPV4_ADDRESS:4001] endpoint health
```

2. Get the crashed master **etcd** member ID by checking the **etcd** cluster status:

```
$K8S_HOME/bin/etcdctl --cacert=$K8S_HOME/ssl/ca.crt --cert=$K8S_
HOME/ssl/server.crt --key=$K8S_HOME/ssl/server.key --endpoints=[https://HA_
VIRTUAL_IP:4001] -w table member list
```

3. Remove the unhealthy **etcd** member corresponding to the crashed master.

```
$K8S_HOME/bin/etcdc --cacert=$K8S_HOME/ssl/ca.crt --cert=$K8S_
HOME/ssl/server.crt --key=$K8S_HOME/ssl/server.key --endpoints=
[https://HA_VIRTUAL_IP:4001] member remove <member>
```

4. Provision a new host with the same IP address as the original one. Run the installation as described in the Transformation Hub Deployment Guide. If the crashed node was labeled originally (for example, as a worker node), then log in to label the new one with the same labels. Refer to the Transformation Hub Deployment Guide for details on labeling nodes.

Pushing JKS files from ArcMC

You can push JKS (Java Keystore) files to multiple managed SmartConnectors in ArcMC. First, you will upload the files to a file repository in ArcMC, and then push them out to their destination SmartConnectors. You must then configure and enable the Kafka destination on all SmartConnectors.

To upload the Java Keystore files:

1. Prepare the .jks files you want to push and store them in a secure network location.
2. In ArcMC, click **Administration > Respositories > New Repository.**
3. In **Name, Display Name**, and **Item Display Name**, enter KAFKA_JKS
4. Enter other required details as needed, and then click **Save**.
5. Click **Upload to Repository.**
6. Follow the prompts in the upload wizard and browse to the first .jks file. Note: make sure to choose the individual file option.
7. Upload as many files as needed by repeating the upload wizard.

To push the files to multiple SmartConnectors:

1. In ArcMC, browse to the file repository for the `.jks` files.
2. Click the **Upload** arrow.
3. Follow the prompts in the wizard and select your destination SmartConnectors.
4. The files are pushed to the managed SmartConnectors and stored in the designated SmartConnector folder.

To configure the Kafka destination on all SmartConnectors:

In ArcMC, click **Node Management > Connectors** tab.

1. Select the SmartConnectors to be configured.
2. Choose **Add a destination** and pick the Kafka destination type.
3. Add the destination details along with the `.jks` path and password, and save the changes.

Liveness Probes

A ***liveness probe*** is a Kubernetes feature that can be configured to detect problematic pods. Once detected, Kubernetes will take action to restart a problematic pod. Liveness probes help ensure higher availability of pods as well as a more robust cluster environment. Consult the Kubernetes documentation for a more detailed explanation of liveness probes.

Transformation Hub supports these liveness probe types:

- TCP/IP port-socket connection
- HTTP request
- Log scanning

Each container or pod supports the listed liveness probes, with their default parameter values shown.

Container/Pod	Probe	initialDelaySeconds	periodSeconds	timeoutSeconds	failureThreshold
Kafka	tcp socket :9092 and log scanning	240	60	30	3
Zookeeper	tcp socket :2181 and log scanning	240	60	30	3
Web Service	https GET :8080 and log scanning	240	300	30	3
Schema Registry	https GET :8081 config and log scanning	240	300	30	3
Kafka Manager	http GET :9000 and log scanning	240	600	30	3
Routing Processor	log scanning	240	60	30	3
C2AV (CEF-to-Avro) Processor	log scanning	240	60	30	3

Probe parameters are defined as follows:

Parameter	Definition
initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated. Note that the first probe execution after startup is not until initialDelaySeconds + periodSeconds.
periodSeconds	How often to perform the probe.
timeoutSeconds	Number of seconds after which the probe times out.
failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up and restarting the pod.

Managing Liveness Probes

To check if a pod has a liveness probe configured:

1. Run:
   `kubectl -n <namespace> describe pod <podname>`
2. Review the output. Look (or `grep`) for the line starting with the string `Liveness...` This will show some of the probe's configuration.

To check for probe failures:

1. Run:
   `kubectl get pods --all-namespaces`

2. If any pod shows 1 or more restarts, run:
   `kubectl -n <namespace> describe pod <podname>`
3. Review any list of events at the end of the output. Liveness probe failures will be shown here.

Configuring Liveness Probes

The default values for liveness probes can be overriden by changing the values of the appropriate properties on the Configuration page.

1. Log in to the Management Portal `(https://<ha-address>:5443).`
2. Click **Administration**.
3. Click the **...** (Browse) icon to the right of the main window.
4. From the drop-down, click **Reconfigure**. The post-deployment settings page is displayed.
5. Browse the configuration properties list to find the desired property, and enter the new value.
5. Click **Save**.

Configuring Log Scanning Liveness Probes

Log scanning probes scan the application's output for a match to a configured pattern, such as a known error message. If the pattern is found, the pod is restarted.

In additon to the four parameters described in the table above, log scanning probes have two additional properties:

literal	A literal expression for matching against the application's log output.
regex	A regular expression for matching against the application's log output.

- The ***literal*** property specifies a literal (exact match) search string. If the value matches a portion of the log text, the liveness probe, on its next periodic check, will report a failure and restart the pod.
- The ***regex*** property is similar, except that a regular expression can be specified for the match. This regex must conform to Java regex rules. To specify a regex escape value within the regex, use 2 backslashes to escape it (`\\`).
- Multiple search patterns can be specified per property, separated by 4 vertical bars (`||||`). A match on any of the patterns will trigger the probe failure.
- There are no default values for these parameters. Log scanning is disabled in the default configuration.
- Matching across multiple rows is not supported. The match must occur on one log line.
- For example, to restart the CEF-to-Avro Routing Stream Processor pod when the value, `Setting stream threads to d` (where `d` could be any single digit), is found in the log, change the configuration property "CEF-to-Avro Routing Stream Processor liveness probes regular expression" to the following value .

```
Setting stream threads to \\d
```

Verification

To verify that log scanning is configured as intended, review the pod's log and look for entries containing `InputStreamScanner`.

For example, to view the c2av-processor pod log, run:

```
kubectl -n <namespace> logs th-c2av-processor-0 | more
```

For the previous property example, the corresponding log line would be:

```
InputStreamScanner: Will scan for RegEx pattern [Setting stream threads to \d]
```

Chapter 5: Managing Transformation Hub Topics

You can manage your Transformation Hub topics through Transformation Hub Manager or through ArcMC.

This section includes the following topics:

Default Topics

Transformation Hub manages the distribution of events to topics, to which consumers can subscribe and receive events from.

Transformation Hub includes the following default topics:

Topic Name	Event Type	Valid Destinations
th-cef	CEF event data.	Can be configured as SmartConnector or Connector in Transformation Hub (CTH) destination.
th-binary_esm	Binary security events, which is the format consumed by ArcSight ESM.	Can be configured as a SmartConnector destination.
th-syslog	The Connector in Transformation Hub (CTH) feature sends raw syslog data to this topic using a Collector.	Should only be configured as Collector or CTH destination.
th-cef-other	CEF event data destined for a non-ArcSight subscriber.	
th-arcsight-avro-sp_metrics	For ArcSight product use only. Routing stream processor operational metrics data.	
th-arcsight-avro	For ArcSight product use only. Event data in Avro format for use by ArcSight Investigate.	
th-arcsight-json-datastore	For ArcSight product use only. Event data in JSON format for use by ArcSight infrastructure management.	

In addition, using ArcSight Management Center, you can create new custom topics to which your SmartConnectors can connect and send events.

Topic Data Preservation

Topic data is preserved across Transformation Hub restarts, reinstalls, and upgrades.

- When a Transformation Hub reinstall is performed, all data in a Kafka topic is preserved. No data is lost.
- When the consumer resumes data collection from the topics, the consumer re-starts where it last left off. No data is lost.

Data Redundancy and Topic Replication

When setting up a Transformation Hub, you can specify the number of copies (replicas) of each topic Transformation Hub should distribute.

Kafka automatically distributes each event in a topic to the number of broker nodes indicated by the topic replication level specified during the Transformation Hub configuration. While replication does decrease throughput slightly, ArcSight recommends that you configure a replication factor of at least 2. You need at least one node for each replica. For example, a topic replication level of 5 requires at least five nodes; one replica would be stored on each node. The following table illustrates how the replication factor provides redundancy in case of unavailable nodes.

Replication Factor	Number of brokers receiving the event	If one node becomes unavailable...
1	1	Data is lost
2 (or more)	Same as replication factor	• Copies of the event data is still present on other node. • Data is restored to an unavailable node when it becomes available again. • No data is lost unless all nodes become unavailable simultaneously.

When you add new consumers, you don't need to update your producers. Transformation Hub handles the distribution and replication for you. Refer to the Kafka documentation for more information.

Managing Topics through ArcMC

You can use ArcMC to view and create topics , as well as to create ***routes***, which direct events into appropriate topics.

A ***route*** is a rule that directs Transformation Hub to duplicate events that meet certain criteria from a source topic to the route's destination topic. Rules are defined using event field names and expected values. Only CEF text format events can be routed. Binary security events in the `th-binary_esm` topic cannot be routed.

Using ArcMC, you can view, create, edit and delete routes based on CEF fields and event metadata. (You must create destination topics before you can route events to them.) Refer to the ArcMC Administrator's Guide, available from the Micro Focus support community, for more information.

Routing Stream Processor Groups

Each source topic is processed by a ***routing stream processor group***. A routing processor group can be scaled independently as load increases by adding more routing processor instances to the group.

Note: Multiple routes using the same source topics will be handled by the same single routing stream processor group for that source topic.

The number of source topics that can be supported is limited by the number of routing stream processor groups (10) that can be configured on the CDF user interface.

Transforming Routing Stream Processors (C2AV)

All instances of Transforming Routing Stream Processors (C2AV processors) created will belong to the same group and they all work together to transform events from the predefined single source (`th-cef`) and target (`th-arcsight-avro`) topics.

The number of routing stream processor instances than can be added to the group is only limited by the system resources. Please add additional instances to the group only if required to meet event processing performance requirements or to improve event processing performance.

Tuning Routing Stream Processors

Routing stream processors and their management are critical to Transformation Hub performance. In general, you can follow these guidelines for tuning stream processors and drive better performance.

- Since all routes which use the same source topic share the same routing stream processor group, adding source topics can speed up processing.
- Increase the number of source topic partitions to handle high EPS throughput, depending on the CPU and memory resources of each worker node. For example, when the partition number equals 60, up to 10 routing (or C2AV) process instances can be used. Each stream process use 6 threads by default.
- Where possible, limit the number of rules per route.

Appendix A: Kubernetes Command Reference

The following Kubernetes commands are commonly used for the operation and administration of Transformation Hub.

Where specified, the string `xxxx` is a variable and its value depends on your specific installation architecture. The actual value can be obtained from the output of the command `get nodes`.

Task	Command
Display all nodes in the cluster	`# kubectl get nodes --all-namespaces`
Display all pods in the cluster	`# kubectl get pods --all-namespaces`
List the nodes with a specific label applied, for example "kafka"	`# kubectl get nodes -L kafka`
Find status of the Transformation Hub application modules	`# kubectl get pods -n arcsight-installer-xxxx -o wide`
Find the pod name for a specific module, like kafka-manager	`# kubectl get pods -n arcsight-installer-xxxx | grep kafka-manager`

Task	Command
Get logs from a specific module, such as web service	Add `-c (container)` parameter if the pod contains more than one container. For example: `# kubectl logs -n arcsight-installer-xxxx th-web-service-{unique ID} -c atlas-web-service`
Execute a single command inside one of the application pods	`# kubectl exec th-c2av-processor-0 -n arcsight-installer-xxxx -- env`

Open a bash shell first, to run multiple commands inside	`# kubectl exec th-c2av-processor-0 -n arcsight-installer-xxxx -it bin/bash`
	`[th-c2av-processor-0 /]#cat /th/sp/config/stream.properties | grep num.stream.threads`
	`num.stream.threads=6`

Glossary

C

Cluster

A group of nodes, pods, or hosts.

Common Event Format (CEF)

CEF is a unified format that transforms log file data into normalized, enriched, and categorized log data. It is an IT security industry standard log format.

Connectors in Transformation Hub (CTH)

CTH features enable enriching, normalizing and sending syslog data and routing it to Kafka topics.

Consumer

A consumer of Transformation Hub event data. Consumers may be Micro Focus products such as Logger or ESM, third-party products like Hadoop, or can be made by customers for their own use.

Container Deployment Foundation (CDF)

CDF is the container-based delivery and management model built on Kubernetes and Docker containers, which standardizes distribution, installation, upgrade, and operation of Micro Focus products and product suites.

CTH

Collector in Transformation Hub (CTH). A feature where SmartConnector technology operates directly in Transformation Hub to collect data.

D

Dedicated Master Node

A node dedicated to running the Transformation Hub Kubernetes control plane functionality only.

Destination

In Micro Focus products, a forwarding location for event data. A Transformation Hub topic is one example of a destination.

Docker container

A Docker container is portable application package running on the Docker software development platform. Containers are portable among any system running the Linux operating system.

F

flannel

flannel (spelled with a lower-case f) is a virtual network that gives a subnet to each host for use with container runtimes. Platforms like Google's Kubernetes assume that each container (pod) has a unique, routable IP inside the cluster. The advantage of this model is that it reduces the complexity of doing port mapping.

Fully Qualified Domain Name (FQDN)

A fully qualified domain name (FQDN) is the complete domain name for a specific computer, or host, on the internet. The FQDN consists of two parts: the hostname and the domain name. For example, an FQDN for a hypothetical mail server might be mymail.example.com. The hostname is mymail, and the host is located within the domain example.com.

I

Initial Master Node

The Master Node that has been designated as the primary Master Node in the cluster. It is from this node that you will install the cluster infrastructure.

K

Kafka

An open-source messaging system that publishes messages for subscribers to consume on its scalable platform built to run on servers. It is commonly referred to as a message broker.

Kubernetes

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery.

L

Labeling

Adding a Kubernetes label to a Master or Worker Node creates an affinity for the workload to the Master or Worker Node, enabling the node to run the specified workload on the labeled server.

Local Docker Registry

The Docker Registry location on the Master and Worker Nodes in the Transformation Hub cluster. Transformation Hub software is launched and managed from the Local Docker Registry.

M

Master Nodes

Master Nodes run the CDF Installer and process web services calls made to Transformation Hub. They connect to, and are administered by, the ArcSight Management Center. A minimum of 1 Master Node is required for each TH cluster.

N

Network File System (NFS)

This is the location where the CDF Installer, Transformation Hub, and other components may store persistent data. A customer-provisioned NFS is required. This environment is referred to in this documentation as an "external" NFS. Although the CDF platform can host a CDF-provisioned NFS (Internal NFS), for high availability an External NFS service should implemented.

Node

A processing location. In Transformation Hub and other containerized applications, nodes come in two types: master and worker.

P

Pod

Applications running in Kubernetes are defined as “pods”, which group containerized components. Transformation Hub uses Docker Containers as these components. A pod consists of one or more containers that are guaranteed to be co-located on the host server and can share resources. Each pod in Kubernetes is assigned a unique IP address within the cluster, allowing applications to use ports without the risk of conflict.

Producer

A gatherer of event data, such as a SmartConnector or CTH. Typically data from a producer is forwarded to a destination such as a Transformation Hub topic.

R

Root Installation Folder

The root installation folder is the top level folder that the Transformation Hub, CDF Installer and all supporting product files will be installed into. The default setting is /opt/arcsight. It is referred to as RootFolder in this document, supporting scripts, and installation materials.

S

Shared Master and Worker Nodes

A configuration where both Master and Worker Nodes reside on the same hosts. This is not a recommended architecture for high availabillty.

SmartConnector

SmartConnectors automate the process of collecting and managing logs from any device and in any format.

T

Thinpool

Using thin provisioning in Docker, you can manage a storage pool of free space, known as a thinpool, which can be allocated to an arbitrary number of devices when needed by applications.

Transformation Hub

A Kafka-based messaging service that enriches and transforms security data from producers and routes this data to consumers.

Transformation Hub cluster

The Transformation Hub cluster consists of all Master and Worker Nodes in the TH environment.

V

Virtual IP (VIP)

To support high availability, a VIP is used as the single IP address or FQDN to connect to a dedicated Master infrastructure that contains 3 or more master Nodes. The Master Nodes manage Worker Nodes. The FQDN of the VIP can also be used to connect to the cluster's Master Nodes.

W

Worker Nodes

Worker nodes ingest, enrich and route events from event producers to event consumers. Worker nodes are automatically load-balanced by the TH infrastructure.

Z

ZooKeeper

In Kafka, a centralized service used to maintain naming and configuration data and to provide flexible and robust synchronization within distributed systems.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Administrator's Guide (Transformation Hub 3.2.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to arcsight_doc@microfocus.com.

We appreciate your feedback!