



# Micro Focus Transformation Hub

Software Version: 3.3.0

## Administrator's Guide

Document Release Date: July 31, 2020

Software Release Date: July 31, 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

© Copyright 2020 Micro Focus or one of its affiliates.

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.

UNIX® is a registered trademark of The Open Group.

## 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

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

# Contents

Chapter 1: Transformation Hub .....	6
Security Open Data Platform (SODP) .....	7
Management Center (ArcMC) .....	7
SmartConnectors .....	8
Logger .....	8
ESM (Enterprise Security Manager) .....	8
Recon .....	9
Chapter 2: Producing and Consuming Event Data .....	10
Producing Events with SmartConnectors .....	10
Consuming Events with ESM .....	11
Consuming Events with Logger .....	11
Sending Transformation Hub Data to Logger .....	12
Example Setup with Multiple Loggers in a Pool .....	14
Consuming Events with Third-Party Applications .....	14
Consuming Transformation Hub Events with Apache Hadoop .....	15
Architecture for Kafka to Hadoop Data Transfer .....	15
Using Apache Flume to Transfer Events to Hadoop .....	15
Setting Up Flume to Connect with Hadoop .....	16
Sample Flume Configuration File .....	16
Setting Up Hadoop .....	18
Connectors in Transformation Hub (CTH) .....	19
Configuring Consumers and Producers for Availability .....	21
Chapter 3: Securing your Transformation Hub deployment .....	22
Changing Transformation Hub Security Mode .....	22
Chapter 4: Managing Transformation Hub .....	24
Licensing Transformation Hub .....	24
Managing Transformation Hub through ArcMC .....	25
About the Transformation Hub Kafka Manager .....	26
Connecting to the Transformation Hub Kafka Manager .....	26
Managing Clusters .....	27
Viewing Information About a Cluster .....	28
Managing Brokers .....	28
Viewing Broker Details .....	29
Summary .....	30
Metrics .....	30

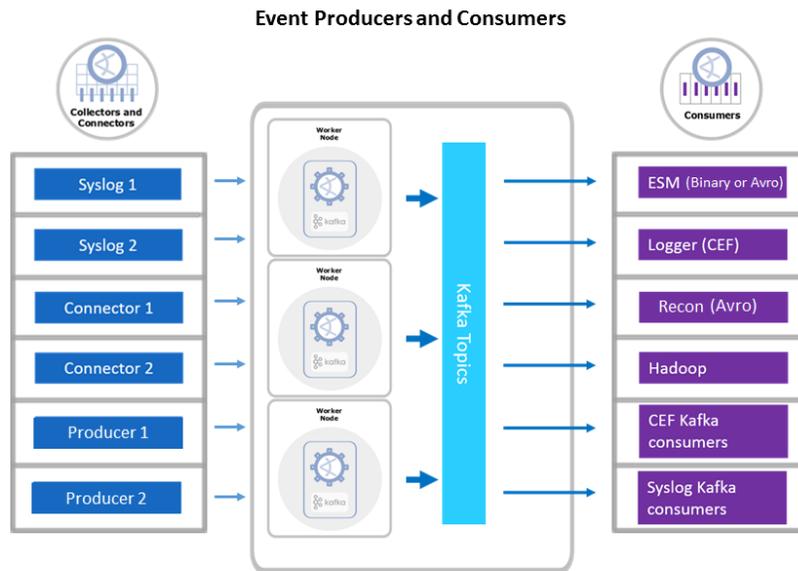
Messages count .....	30
Per Topic Detail .....	30
Managing Topics .....	30
Creating Topics .....	31
Viewing Topic Details .....	32
Topic Summary .....	32
Metrics .....	32
Operations .....	32
Partitions by Broker .....	33
Consumers consuming from this topic .....	33
Partition Information .....	34
Managing Consumers .....	34
Viewing Consumer Details .....	34
Data Compression .....	35
Managing Preferred Replicas .....	36
Managing Partitions .....	37
Configuring Topic Partitions Based on Number of Consumers .....	37
Graceful Shutdown and Rebooting of Transformation Hub Nodes (On-Premises Only) .....	38
Adding a New Worker Node .....	38
Adding ZooKeeper Instances .....	39
Adding a Kafka Broker .....	40
Uninstalling a Master or Worker Node .....	41
Removing a Crashed Worker Node .....	42
Replacing a Crashed Master Node .....	42
Pushing JKS files from ArcMC .....	43
Liveness Probes .....	44
Chapter 5: Managing Transformation Hub Topics .....	47
Data Redundancy and Topic Replication .....	47
Creating Routes for Topics .....	48
Stream Processor Groups .....	48
Filtered Avro Events for ESM .....	49
Appendix A: Kubernetes Command Reference .....	50
Appendix B: Overriding Application Properties .....	52
Send Documentation Feedback .....	55
Glossary .....	56

# Chapter 1: Transformation Hub

Transformation Hub, whether deployed on-premises or in an Azure environment, is the high-performance message bus for ArcSight 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 Recon to receive the event stream, while smoothing event spikes, and functioning as an extended cache.

Transformation Hub ingests, enriches, normalizes, and then routes event data from data producers to connections between existing data lakes, analytics platforms, and other security technologies and the multiple systems within the Security Operations Center (SOC). 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 Recon technologies to push event data to HDFS for long-term, low-cost storage. Recon is integrated with the Transformation Hub for Avro events. ESM receives binary and Avro event data for dashboarding and further correlation.

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



## Security Open Data Platform (SODP)

The Security 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)
- SmartConnectors

## Management Center (ArcMC)

The ArcMC management console enables central administration of 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 ArcSight systems 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

ArcSight 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 include:

- 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

## ESM (Enterprise Security Manager)

Enterprise Security Manager (ESM) is a comprehensive software solution that combines traditional security event monitoring with network intelligence, context correlation, anomaly detection, historical analysis tools, and automated remediation. ESM is a multi-

level solution that provides tools for network security analysts, system administrators, and business users.

ESM includes the Correlation Optimized Retention and Retrieval (CORR) Engine, a proprietary data storage and retrieval framework that receives and processes events at high rates, and performs high-speed searches.

## Recon

ArcSight Recon is a comprehensive log management and search solution that eases compliance burdens and accelerates forensic investigation for security professionals. It combines the compliance, storage and reporting needs of log management with the capabilities of big-data search and analysis. Recon is built for security event logs and is therefore more intuitive and accessible for security analysts, it won't require a DBA to operate. It helps hunt and defeat threats by unifying data logs from across organizations, processing billions of events, and quickly making them available for search, visualization and reporting.

Recon helps SOC analysts gain a deeper understanding of alerts across their organization and plays an important role in ArcSight's mission to deliver powerful layered analytics.

## 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, Recon, 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](#) ..... 10
- [Consuming Events with ESM](#) ..... 11
- [Consuming Events with Logger](#) ..... 11
- [Consuming Events with Third-Party Applications](#) ..... 14
- [Consuming Transformation Hub Events with Apache Hadoop](#) ..... 15
- [Connectors in Transformation Hub \(CTH\)](#) ..... 19
- [Configuring Consumers and Producers for Availability](#) ..... 21

### 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 Recon, 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.

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 available for download from 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 or Avro format for the topics to which it is subscribed.

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 *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 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 and reverse DNS lookup, and be 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](#) ..... 15
- [Using Apache Flume to Transfer Events to Hadoop](#) ..... 15
- [Setting Up Flume to Connect with Hadoop](#) ..... 16
- [Sample Flume Configuration File](#) ..... 16
- [Setting Up Hadoop](#) ..... 18

## 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 <code>org.apache.flume.source.kafka.KafkaSource</code> .
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](#).

For a detailed discussion of connecting Apache Flume with Hadoop, consult the Apache online documentation here:

<https://flume.apache.org/FlumeUserGuide.html>

## 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 (CTH)

To reduce the computational overhead and workload on a syslog SmartConnector infrastructure, you can make use of Connectors in Transformation Hub (CTH) instead.

### CTH Functionality

Operationally, Micro Focus SmartConnectors hold two main responsibilities:

- **Collection:** A SmartConnector collects data from various sources.
- **Processing:** A SmartConnector processes the collected data into enriched security event data and posts them to a destination.

With CTH, the two functions of SmartConnector are handled in a slightly different manner. CTH takes advantage of the massive scalability of the robust Transformation Hub streaming architecture by moving the computationally intensive processing step directly to Transformation Hub.

- **Collection:** The collection step is performed by a dedicated Collector component, which gathers raw syslog data and publishes it to a dedicated syslog topic in Transformation Hub. As the name suggests, a Collector is a lightweight component responsible solely for collecting syslog data and passing it along to a dedicated CTH topic. A Collector is deployed on a VM or server using ArcMC.
- **Processing:** The CTH component reads the data from the Collector destination, and then parses, normalizes, enriches, and filters this data. It posts the data to a dedicated Transformation Hub topic for availability to any desired consumer. CTHs are deployed as Kubernetes pods within the CDF infrastructure.

CTH includes the majority of the functionality of ArcSight syslog connectors, except for data collection, which is handled by the lightweight Collector component instead. For more information on CTH configuration, consult the [ArcSight Syslog Connector User Guide](#).

### Advantages of CTH

CTH has the following advantages over traditional SmartConnector architecture.

- Hardware consolidation in the data collection layer where Collectors are deployed, due to the logical separation of collection and processing. A single data feed from a

Collector can replace multiple SmartConnector feeds.

- Improved stability, easy horizontal scalability, and improved load balancing as data flows increase with time or fluctuate during operations.
- Ease of deployment, since CTHs are deployed with a single click in the ArcMC management console.
- Raw syslog data is now available in the CTH topic and can be shared with any desired consumer.

## Limitations of CTH

- CTH presently supports the processing of syslog data only.
- Upgrades to CTH are performed by upgrading Transformation Hub, rather than by upgrading CTH itself.

## Deploying and Managing CTH

Installation and management of CTH is performed on a managed Transformation Hub through the ArcMC management console. Consult the [ArcMC Administrator's Guide](#) for instructions on how to deploy and manage CTH.

## Destination Topics

Collectors should only be configured with the `th-syslog` topic as a destination (and no other destinations).

Valid routing topic destinations for CTH include the following:

- `th-cef`
- `th-binary_esm`
- `th-cef-other`

In addition, custom CTH source and destination topics may be configured on Transformation Hub. (Custom topics may only be created for CEF data.)

## Collector/CTH Supported Security Modes

*Collector destinations* can support the following security modes:

- Plain text (no security mode selected)
- FIPS only
- TLS only

Collector security mode can be set during Instant Deployment in the ArcMC console. See the ArcMC Administrator's Guide for more information.

*CTH source and destinations* can support the following security modes:

- TLS + Client Authentication (default setting)
- FIPS + Client Authentication (automatically set when enabling FIPS mode in Transformation Hub).
- Plain text (no security mode selected)
- TLS only
- FIPS only

If desired, CTH's plain text, TLS-only, and FIPS-only modes can be set in ArcMC after deployment.

## 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 SmartConnectors), and the consumers subscribing to data from the Transformation Hub (such as Logger and ESM).

### 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 example, `broker_hostname1:9093, broker_hostname2:9093, broker_hostname3:9093`.

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](#) ..... 22

## 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. A 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.

To protect against unknown clients sending events to Transformation Hub, or changing Avro topic schemas, enabling Client Authentication is recommended.

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.

**Note:** For an Azure cluster, please run all Azure Cloud Shell (command line) and `kubectl` commands from the authorized jump host.

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 (Logger, 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 continue from the last offset later.)
3. Log in to the CDF Management Portal.
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:

• <a href="#">Licensing Transformation Hub</a> .....	24
• <a href="#">Managing Transformation Hub through ArcMC</a> .....	25
• <a href="#">About the Transformation Hub Kafka Manager</a> .....	26
• <a href="#">Managing Clusters</a> .....	27
• <a href="#">Managing Brokers</a> .....	28
• <a href="#">Managing Topics</a> .....	30
• <a href="#">Managing Consumers</a> .....	34
• <a href="#">Data Compression</a> .....	35
• <a href="#">Managing Preferred Replicas</a> .....	36
• <a href="#">Managing Partitions</a> .....	37
• <a href="#">Graceful Shutdown and Rebooting of Transformation Hub Nodes (On-Premises Only)</a> .....	38
• <a href="#">Adding a New Worker Node</a> .....	38
• <a href="#">Adding ZooKeeper Instances</a> .....	39
• <a href="#">Adding a Kafka Broker</a> .....	40
• <a href="#">Uninstalling a Master or Worker Node</a> .....	41
• <a href="#">Removing a Crashed Worker Node</a> .....	42
• <a href="#">Replacing a Crashed Master Node</a> .....	42
• <a href="#">Pushing JKS files from ArcMC</a> .....	43
• <a href="#">Liveness Probes</a> .....	44

### 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 CDF Management Portal.
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 pods from 0 to x, one at a time, 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 before restarting the next pod:  

```
kubectl logs th-kafka-(x) -n arcsight-installer-XXX
```

### 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](#).

**Note:** For an Azure cluster, see the section "Connecting to Kafka Manager" in the chapter "Maintaining an Azure Cluster" of the Transformation Hub Deployment Guide.

### To access Transformation Hub Kafka Manager:

1. On a Transformation Hub node, run the command to get the Kafka 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. In a browser, connect by using the following URL:

```
http://<localhost>:<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 default topics 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`

`mf-event-avro-esmfiltered`

```
mf-event-cef-esmfiltered
```

### 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 `kubect1 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 broker 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 (Recon 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 message.

## 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 .

## Data Compression

Transformation Hub compression settings affect data in two general places, communication and storage. Specifically, this refers to data stored on disk, in Kafka topic partitions, and data that is in transit.

- All external producers such as connectors, collectors, and internal producers, like routing and CEF2Avro processors, compress data before sending it.
- For data in transit, data compression is controlled by the producer's configuration.

## Data Consumers

There is no property that controls data compression on consumers. Consumers read metadata from each message, which indicates the correct decompression algorithm to use. Since this is evaluated on a message-by-message basis, the consumer's behavior does not depend on which topic it is consuming from. A single topic may contain messages which have been compressed with different compression algorithms (also referred to as compression types or codecs).

## Data Storage (Data at Rest)

The algorithm used to compress stored data is determined by the topic configuration. All Transformation Hub topics, except `th-arcsight-avro`, currently use the default compression type, which is the same as that used by `producer`. This configuration choice means the topic will retain the original compression algorithm set by the producer. By leaving this as producer-defined, there is flexibility for the producer to send either compressed (using any supported codec) or uncompressed data.

The `th-arcsight-avro` topic is an exception because the database scheduler reads from this topic, but does not yet have support for reading messages encoded with the ZStandard (`zstd`) compression algorithm. Therefore, there is a specific, out-of-the-box value for this topic, to insure that the database scheduler can read it, no matter what over-the-wire compression was used.

Topic	Compression Type	Transformation Hub Version
All topics except <code>th-arcsight-avro</code>	producer (default)	3.3 and earlier
<code>th-arcsight-avro</code>	<code>gzip</code>	3.3
<code>th-arcsight-avro</code>	uncompressed	3.2 and earlier

## Configuring Compression

There are two places in the Kafka architecture where compression can be configured: the producer and the topic.

- Producer-level compression is set on the producer; for example, in SmartConnector Transformation Hub destination parameters. For producers that reside inside TH, such as routing and stream processors, the compression algorithm is configured on the Transformation Hub configuration page, during deployment.
- Topic-level compression can be set with Kafka Manager (using **Topic > Update Config Menu**); however, it is strongly recommended that settings be left at default values.

## Compression Types

While Kafka supports a handful of compression types, Transformation Hub implements only two types: `gzip` and `zstd`.

- **gzip:** By default, `gzip` is used for Transformation Hub routing and stream processors, as well as for SmartConnectors. This is for backward compatibility and may change in a future release.
- **zstd:** Testing has shown that `zstd` uses less bandwidth, storage, and CPU resources than `gzip`. For bandwidth constrained networks, higher EPS is typically seen when using `zstd`; however actual results are unique to each environment. Third-party Java producers should use `kafka-clients` version 2.1.0 or later, for `zstd` support. In addition to Transformation Hub 3.3.0, ArcSight consumers compatible with `zstd` include Logger 7.0, ESM 7.2, IDI 1.1, or later.

## 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 (On-Premises Only)

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

**Note:** This section does not apply to Azure clusters. For adding new nodes to Azure clusters, see the section "Scaling Up a Cluster" in the Transformation Hub Deployment Guide.

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

### To add a new worker node:

1. Set up and provision the new node according to the guidelines and system requirements given in the *Micro Focus CDF On-Premises 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 command to update the shared volumes:  
`exportfs -ra`
4. Log in to the CDF Management Portal (<https://<ha-address>:5443>).
5. Click **Cluster > Nodes**.
6. Click **+ Add**.
7. Enter values for the pop-up dialog. For host name, use the FQDN of the new node.
8. Click **ADD**.

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

## Adding ZooKeeper Instances

Adding a ZooKeeper instance has the following prerequisites:

- You will need at least 2 available worker nodes already deployed that do not already have a Kafka broker or ZooKeeper instance deployed on them. (Only one ZooKeeper can be installed for each worker node, and there must be an odd number of ZooKeepers --1, 3, 5, 7, and so on. Therefore you need at least 2 additional worker nodes to keep the total number an odd one.)
- Any new node where a ZooKeeper instance is deployed should be labeled `zk=yes`.
- If you plan to deploy Kafka brokers as well, perform this procedure to add ZooKeeper instances *before* you have deployed your Kafka brokers.

### To add new ZooKeeper instances:

1. Open the CDF Management Portal.
2. Click ... (**Browse**) to the right.

3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of ZooKeeper nodes in the ZooKeeper cluster*.
5. From the field's drop-down, select the new number of ZooKeeper instances.
6. Click **Save**.
7. Verify the new ZooKeeper pods are up and in Running state by running the command:  

```
# kubectl get pods -n {arcsight_namespace_id} th-zookeeper-x
```

**Note:** Reducing the number of ZooKeeper instances is not currently supported.

## Adding a Kafka Broker

Adding a new Kafka broker has the following prerequisites:

- You will need an available worker node already deployed that does not already have a Kafka broker or ZooKeeper instance deployed on it.
- The new node where a Kafka broker is deployed should be labeled `kafka=yes`.
- If you plan to deploy ZooKeepers as well, perform this procedure to add Kafka brokers after you have deployed your ZooKeepers and they are up and running.

### To add a Kafka broker:

1. Open the CDF Management Portal.
2. Click ... (**Browse**) to the right.
3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of Kafka broker nodes in the ZooKeeper cluster*.
5. From the field's drop-down, select the new number of Kafka brokers.
6. Adjust any other related fields as needed. For example, if the topic replication factor is 1, consider increasing it.
7. Click **Save**.
8. Verify that the new Kafka brokers are up and in Running state by running the command:  

```
kubectl get pods -n {arcsight_namespace_id} th-kafka-x
```

### Next, to assign partitions to a new Kafka broker:

1. Connect to Transformation Hub Kafka Manager.

**Note:** Refer to [Connecting to the Transformation Hub Kafka Manager](#) for more information.

2. In **Cluster > Transformation Hub > Topics**, click **Generate Partition Assignments**.
3. On the **Confirm Assignments** page, confirm partition assignments for the new broker and click **Generate Partition Assignments**.
4. On the main toolbar, click **Topic > List**.
5. Click **Run Partition Assignments**.
6. On the **Run Assignments** page, confirm partition assignments and click **Run Partition Assignments**.
7. 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.

**Note:** Reducing the number of Kafka brokers is not currently supported.

## 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

**Note:** This section does not apply to Azure clusters, since Azure clusters do not contain master nodes.

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_FQDN:4001,https://MASTER2_FQDN:4001,https://MASTER3_FQDN: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.

```
$K8S_HOME/bin/etcd --cacert=$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 > Repositories > 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
<b>Kafka</b>	tcp socket :9092 and log scanning	240	60	30	3
<b>Zookeeper</b>	tcp socket :2181 and log scanning	240	60	30	3
<b>Web Service</b>	https GET :8080 and log scanning	240	300	30	3
<b>Schema Registry</b>	https GET :8081 config and log scanning	240	300	30	3
<b>Kafka Manager</b>	http GET :9000 and log scanning	240	600	30	3
<b>Routing Processor</b>	log scanning	240	60	30	3
<b>C2AV (CEF-to-Avro) Processor</b>	log scanning	240	60	30	3

Probe parameters are defined as follows:

Parameter	Definition
<b>initialDelaySeconds</b>	Number of seconds after the container has started before liveness probes are initiated.  Note that the first probe execution after startup is not until <code>initialDelaySeconds + periodSeconds</code> .
<b>periodSeconds</b>	How often to perform the probe.
<b>timeoutSeconds</b>	Number of seconds after which the probe times out.
<b>failureThreshold</b>	When a Pod starts and the probe fails, Kubernetes will try <code>failureThreshold</code> 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 overridden by changing the values of the appropriate properties on the Configuration page.

1. Log in to the CDF Management Portal.
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 addition to the four parameters described in the table above, log scanning probes have two additional properties:

<b>literal</b>	A literal expression for matching against the application's log output.
<b>regex</b>	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:

- [Data Redundancy and Topic Replication](#) .....47
- [Creating Routes for Topics](#) .....48
- [Stream Processor Groups](#) .....48
- [Filtered Avro Events for ESM](#) .....49

## 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	<ul style="list-style-type: none"> <li>• Copies of the event data is still present on other node.</li> <li>• Data is restored to an unavailable node when it becomes available again.</li> <li>• No data is lost unless all nodes become unavailable simultaneously.</li> </ul>

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.

## Creating Routes for Topics

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 (filter) 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.

## Stream Processor Groups

Transformation Hub implements two types of stream processors to process events:

- *Routing stream processors* process CEF event data and send it to CEF destinations, based on Transformation Hub routing rules specified in ArcSight Management Center. This data remains in CEF format.
- *Transforming stream processors* (also called *CEF to Avro* or *C2AV*) work to transform CEF data in the CEF source topic and route it to the dedicated Avro destination topic (`th-arcsight-avro`), for use by Avro consumers by default. There are two C2AV stream processor types, each working with a dedicated set of topics.
  - The CEF-to-Avro stream processor transforms events from the `th-cef` topic to the `th-arcsight-avro` topic.
  - New in Transformation Hub 3.3, the CEF-to-Avro ESM Filtered Stream Processor transforms events from the `mf-event-cef-esmfiltered` topic to the `mf-event-avro-esmfiltered` topic.

Each stream processor comprises 6 processing threads.

All routes with the same source topic are processed by one *routing stream processor group*. A processor group can be scaled independently as load increases by adding more routing processor instances to the group. The number of routing stream processor groups should match the number of source topics they are processing. Each routing stream processor group can contain multiple routing stream processors.

- You can configure up to 10 routing stream processor groups on your Transformation Hub in the CDF Management Portal; so 10 is the limiting factor for the number of

source topics that Transformation Hub can support.

- Transformation Hub ships with two dedicated C2AV processing groups.

Routing is configured in the ArcSight Management Center; see the Administrator's Guide for that product for more information.

## Tuning Stream Processor Groups

The performance of stream processors is 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 more 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 is increased to 60, up to 10 routing (or C2AV) process instances can be used. Each stream processor uses 6 threads by default.
- Where possible, limit the number of routing rules per route.

## Filtered Avro Events for ESM

By combining routing and stream processing, CEF events can be filtered and converted to Avro events for consumption by ESM.

Transformation Hub enables routing, filtering and transformation of CEF events. CEF events can be converted to Avro event format by C2AV Stream Processors in Transformation Hub. Converting Avro events for consumption by ESM is a 2-part process. First, the CEF events must be filtered, and then Transformation Hub can convert those filtered CEF events to Avro events.

**1. Create the filtered CEF events:** In ArcMC, create a route with `th-cef` as the source topic and `mf-event-cef-esmfiltered` as the destination topic. For more information, please refer to the routing section in the ArcMC Administration Guide.

**2. Create the Avro events from the filtered CEF events:** In Transformation Hub, open the CDF Management Portal and browse to the **Post-Configuration Settings** page. Enable the CEF-to-Avro ESM Filtered stream processor by adjusting the running stream processor instance number on Transformation Hub to a number greater than 0. For more information on adjusting post-configuration settings, see the Post-Configuration section of the Transformation Hub Deployment Guide.

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

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: <pre># kubectl logs -n arcsight-installer-xxxx th-web-service-{unique ID} -c atlas-web-service</pre>
Execute a single command inside one of the application pods	<pre># kubectl exec th-c2av-processor-0 -n arcsight-installer-xxxx -- env</pre>
Open a bash shell first, to run multiple commands inside	<pre># kubectl exec th-c2av-processor-0 -n arcsight-installer-xxxx -it bin/bash</pre> <pre>[th-c2av-processor-0 /]#cat /th/sp/config/stream.properties   grep num.stream.threads</pre> <pre>num.stream.threads=6</pre>

## Appendix B: Overriding Application Properties

Each Transformation Hub module (Kafka, Zookeeper, and so on) has many additional properties available, and there may be a need for system administrators to override the default values for some of these properties. This section covers how to override these property values.

Property values (for properties that support overrides) are set by injecting environment variables in the respective container's start-up environment. These variables are read from a user-supplied properties file, in a specific location on the Network File Server (NFS). To see the available properties for override, consult the respective module's published documentation.

- For Kafka, ZooKeeper, and Schema Registry properties, consult the Confluent documentation.
- The properties for routing processor and stream processor modules are detailed below.

**Note:** In most cases, this feature is not required for normal operation of Transformation Hub, and most likely will be used at the direction of technical support. Not all properties support overrides; please check with technical support before making any changes to your configuration.

### Steps to configure

1. Create a file named `<NFS Volume mount>/transformationhub/config/arcsight-env-override.properties`
2. Add properties to the file. To each property, add the module prefix from the table below.

Module	Prefix
Kafka	<code>arcsight.eventbroker.kafka.</code>
Schema Registry	<code>arcsight.eventbroker.schema-registry.</code>
ZooKeeper	<code>arcsight.eventbroker.zookeeper.</code>
Routing Processor/Stream Processor	<code>arcsight.eventbroker.sp.</code>

3. Delete the pods for which properties were defined, or, alternatively, redeploy Transformation Hub.
4. To verify the changes, search the log file (after the container's status is back to Running) for matching properties.

## Routing Processor and Stream Processor Properties

As explained above, prefix these properties with `arcsight.eventbroker.sp.` to create an override.

Property Name	Default Value	Description
RETRIES	2147483647	The number of retries for broker requests that return a retry-able error.
RETRY_BACKOFF_MS	100	The amount of time (milliseconds), before a request is retried. This applies if the retries parameter is configured to be greater than 0.
RECEIVE_BUFFER_BYTES	65536	The size of the TCP receive buffer to use when reading data. If the value is -1, the OS default will be used.
MAX_PARTITION_FETCH_BYTES	1048576	The maximum amount of data per-partition the server will return. Records are fetched in batches by the consumer.
MAX_REQUEST_SIZE	1048576	The maximum size of a request in bytes.
BUFFER_MEMORY	33554432	The total bytes of memory the producer can use to buffer records waiting to be sent to the server.
BATCH_SIZE	16384	the default batch size in bytes when batching multiple records sent to a partition
LINGER_MS	100	the producer will wait for up to the given delay to allow other records to be sent so that the sends can be batched together
HEARTBEAT_INTERVAL_MS	1000	The expected time (milliseconds) between heartbeats to the consumer coordinator when using Kafka's group management facilities. Heartbeats are used to ensure that the consumer's session stays active and to facilitate rebalancing when new consumers join or leave the group.
MAX_POLL_INTERVAL_MS	3600000	The maximum delay (milliseconds) between invocations of poll() when using consumer group management
MAX_POLL_RECORDS	100	The maximum number of records returned in a single call to poll().
SESSION_TIMEOUT_MS	180000	The timeout (milliseconds) used to detect client failures when using Kafka's group management facility
REQUEST_TIMEOUT_MS	305000	The configuration controls the maximum amount of time (milliseconds) the client will wait for the response of a request.
CONNECTIONS_MAX_IDLE_MS	540000	The maximum amount of time (milliseconds) before idle connections are closed.
NUM_STREAM_THREADS	6	The number of threads to execute stream processing.

## Example

To change the value of `ZOOKEEPER_MAX_CLIENT_CNXNS` to 65, in ZooKeeper, and to change the value of `SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS` in the Schema Registry, create a file, `<NFS Volume mount>/transformationhub/config/arcsight-env-override.properties`, and add the following lines:

```
arcsight.eventbroker.zookeeper.ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

```
arcsight.eventbroker.schema-registry.SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS=20000
```

Example of verifying the change by searching the log:

```
# kubectl -n transformationhub1 logs eb-zookeeper-0 | grep ZOOKEEPER_MAX_CLIENT_CNXNS
Environment override script set: ZOOKEEPER_MAX_CLIENT_CNXNS=65
ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

# 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 3.3.0)**

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](mailto:arcsight_doc@microfocus.com).

We appreciate your feedback!

# 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 be 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 availability.

### **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.