Hewlett Packard
Enterprise

HPE Security ArcSight Data Platform Event Broker

Software Version: 2.11

Administrator's Guide

February 9, 2018

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise 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. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

The network information used in the examples in this document (including IP addresses and hostnames) is for illustration purposes only.

HPE Security ArcSight products are highly flexible and function as you configure them. The accessibility, integrity, and confidentiality of your data is your responsibility. Implement a comprehensive security strategy and follow good security practices.

This document is confidential.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise 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

Follow this link to see a complete statement of copyrights and acknowledgements:
https://www.protect724.hpe.com/docs/DOC-13026

Support

Contact Information

Phone	A list of phone numbers is available on the HPE Security ArcSight Technical Support Page: https://softwaresupport.hpe.com/documents/10180/14684/esp-support-contact-list
Support Web Site	https://softwaresupport.hpe.com
Protect 724 Community	https://www.protect724.hpe.com

Contents

Chapter 1: Overview 6

About ADP Event Broker 6

Event Broker Benefits 6

New Features and Enhancements 7

Event Broker architecture 8

Setting up the ADP Event Broker in your environment 9

Deploying Event Broker 9

Chapter 2: Producing and Consuming Event Data 11

Producing Events with SmartConnectors 11

Pushing JKS files from ArcMC 12

Consuming Events with ArcSight Investigate & HPE Vertica 13

Consuming Events with ESM 13

Consuming Events with Logger 13

Sending Event Broker Data to Logger 14

Example Setup with Multiple Loggers in a Pool 15

Consuming Events with Non-ArcSight Applications 16

Using Apache Flume to Transfer Events 16

Consuming Event Broker Events with Apache Hadoop 17

Architecture for Kafka to Hadoop Data Transfer 18

Setting Up Flume to Connect with Hadoop 18

Sample Flume Configuration File 19

Setting Up Hadoop 21

Connectors in Event Broker (CEB) 22

Avoiding Single Points of Failure 22

Event Broker Sizing 23

Chapter 3: Securing Your Event Broker Deployment 25

Firewall Configuration 25

Changing Event Broker security mode 25

Chapter 4: Managing Event Broker 27

Managing Event Broker through ArcMC ... 27
Enabling Event Broker management through ArcMC ... 27
About the Event Broker Manager ... 27
Connecting to the Event Broker Manager ... 28
Managing Clusters ... 29
Viewing Information About a Cluster ... 29
Managing Brokers ... 30
Viewing Broker Details ... 30
Summary ... 31
Metrics ... 31
Messages count ... 31
Per Topic Detail ... 31
Managing Topics ... 31
Creating Topics ... 32
Viewing Topic Details ... 33
Topic Summary ... 33
Metrics ... 33
Operations ... 33
Partitions by Broker ... 34
Consumers consuming from this topic ... 35
Partition Information ... 35
Managing Consumers ... 35
Viewing Consumer Details ... 36
Managing Preferred Replicas ... 36
Managing Partitions ... 36
Command Reference ... 37
Graceful Server Reboot ... 38

Chapter 5: Managing Event Broker Topics ... 40
Default topics ... 40
Topic Configuration ... 40
Data redundancy and topic replication ... 41
Managing topics through ArcMC ... 41

Chapter 6: Licensing ... 43
Applying a New License File ... 43
How to check the license log ... 43
License Check Behavior ... 44

Chapter 7: Troubleshooting 47
Diagnostic Data and Tools 47
Verifying the health of the Event Broker cluster 48
Master or Worker Nodes Down 49
Diagnosing Common Event Broker Issues 50
Event Broker Cluster Down 51
Pod Start Order 51
Cannot query ZooKeeper 51
Common Errors and Warnings in ZooKeeper logs 52
Common Errors and Warnings in Kafka logs 52
Tuning Event Broker Performance 54
Increasing Stream Processor EPS 54
Increasing Kafka retention size/time 54
Adding a new worker node 54

Glossary 57

Send Documentation Feedback 61

Chapter 1: Overview

This chapter includes the following topics:

- About ADP Event Broker 6
- New Features and Enhancements 7
- Event Broker architecture 8
- Setting up the ADP Event Broker in your environment 9
- Deploying Event Broker 9

About ADP Event Broker

The ArcSight Data Platform Event Broker centralizes event processing, enables topic sorting and routing of events, helps you to scale your ArcSight environment, and opens up ArcSight event data to third-party solutions.

It enables you to take advantage of scalable, highly available, multi-broker clusters for publishing and subscribing to event data. The ADP Event Broker integrates with ArcSight Connectors, Logger, and ESM, can be managed and monitored by ArcMC, and is foundational for using ArcSight Investigate.

The ArcSight Data Platform Event Broker provides a packaged version of Apache Kafka. After you install and configure an Event Broker cluster, you can use ADP SmartConnectors to publish data, and subscribe to that data with ADP Logger, ArcSight ESM, ArcSight Investigate, Apache Hadoop, or your own consumer

Event Broker Benefits

In addition to open-source Kafka's distributed, high-performance message buses, with resilient redundant message pipelines, Event Broker has many benefits above and beyond open-source Kafka:

- Event Broker is enterprise-ready with these features:
  - **Container-based deployment:** Makes use of Kubernetes deployment for fast central deployment.
  - **Centralized and local management:** Leverage the central management capabilities of ArcSight Management Center, or manage locally with Event Broker Manager.
  - **System and application monitoring:** Monitor Event Broker metrics like your other ArcSight applications, through ArcSight Management Center.
- Event Broker is optimized for deployment with these features:
  - **Ready-to-go security hardening:** Supports TLS 1.2, FIPS, and Client Authentication.
  - **Event filtering and routing:** Categorize and sort events as needed.

- **Format transformation engine:** Efficiently converts CEF data to Avro format.
- **Ready-to-go topics:** Default topics are installed with the product to get you started quickly.

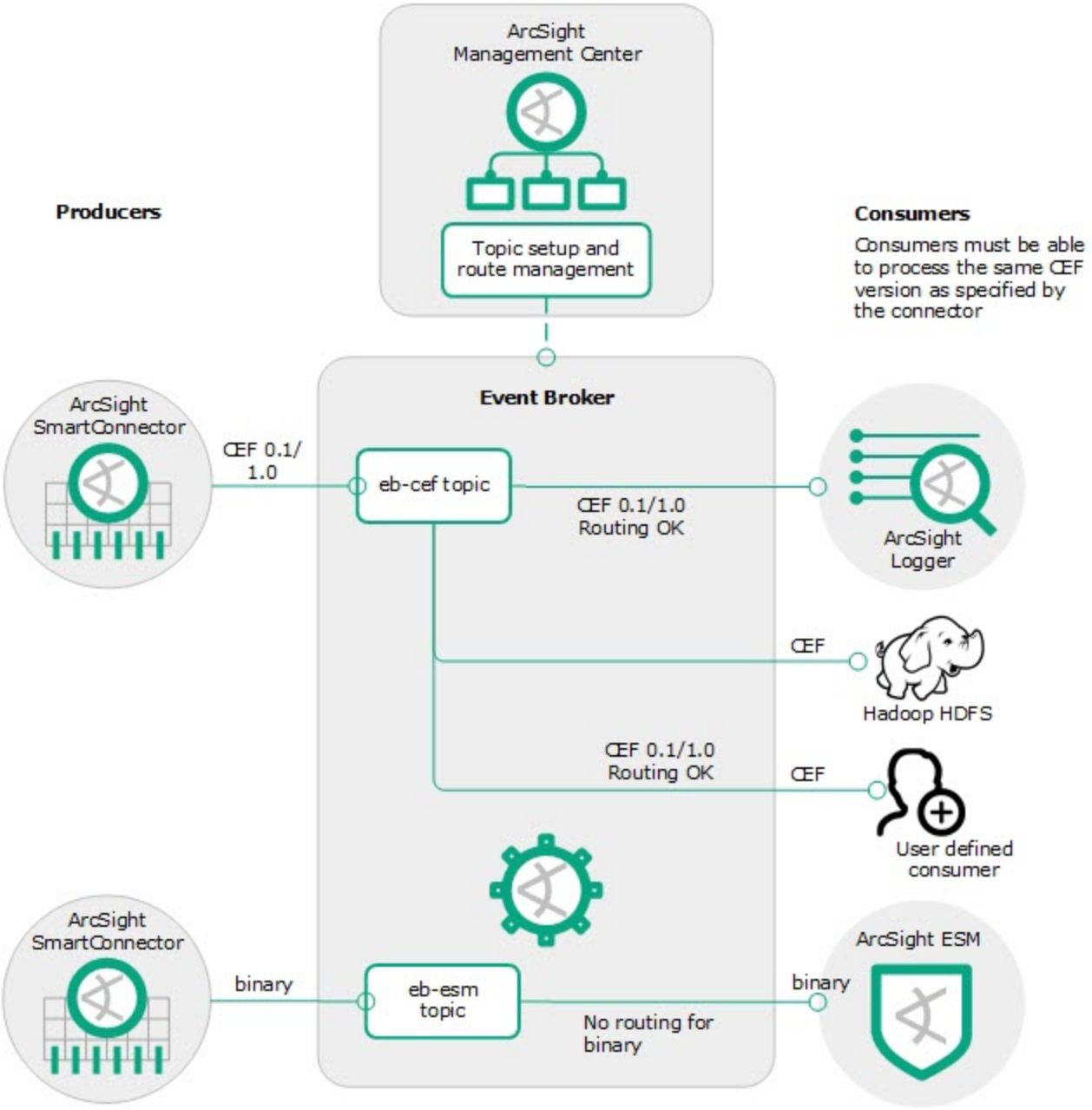
New Features and Enhancements

Event Broker 2.1 includes the following new features and enhancements.

- **TLS 1.2:** Event Broker 2.1 supports TLS 1.2. Note that any consumers or producers that connect to Kafka must use TLS 1.2.
- **Confluent Platform Upgrade:** Event Broker now uses Confluent Platform 3.3.0, which includes Kafka 0.11.0.0.
- **CEB (Evalulation Only):** Event Broker includes support for Connectors in Event Broker (CEB). This new functionality moves the security event normalization, categorization, and enrichment of connectors processing to the Docker containers environment of Event Broker, while reducing the work done by the system component left outside of Event Broker to collection of raw data (the new Collector).

Note: Connectors in Event Broker (CEB) and all related functionality, including Collectors, are provided as **non-production public alpha features.** These features are provided for your testing and evaluation only and should not be considered fully functional, nor are they supported by HPE Support, nor are they guaranteed to be available in the product in the future. Consult the ArcMC Admin Guide, and directions from the ADP product team, for best practices and guidance on how to use these features. **CEB and Collectors should not in any circumstances be used in a production environment.** We welcome questions, comments, and feedback on these features. Please direct any questions or comments to our ADP product team at adp-ceb-alpha@hpe.com.

Event Broker architecture



ArcSight SmartConnectors are the producers that publish data to the ADP Event Broker. Data is sent to Event Broker (in CEF or binary format) to the corresponding topic on the Event Broker.

- You can further route CEF data to specialized topics for categorization.
- Binary data can be used without transformation by ArcSight ESM. Binary topics may not be routed.

Once the data has been processed by Event Broker, you can subscribe to it with producers such as ArcSight Investigate (through Vertica), ArcSight ESM, ADP Logger, Apache HDFS, or your own third-party consumer.

You can manage Event Broker with ArcMC, the ArcSight central management and monitoring solution, and Event Broker Manager.

Setting up the ADP Event Broker in your environment

Set up and test your Event Broker in a staging environment before deploying to a production environment.

The process of setting up Event Broker in your environment includes the following steps:

1. **Review Event Broker Requirements:** Review the Event Broker technical requirements, and prepare your Kubernetes nodes for Event Broker deployment. Refer to the Event Broker Deployment Guide.
2. **Deploy Event Broker:** Deploy Event Broker to the designated Kubernetes nodes, as described in the deployment guide.
3. **Set Up Producers:** Set up one or more SmartConnectors (version 7.6 or later) to produce data for Event Broker. For more information, see the Smart Connectors User Guide.
4. **Install ArcSight Management Center (ArcMC):** If you plan to manage Event Broker with ArcMC (recommended), install your ArcMC. For more information, refer to the ArcMC Administrator's Guide.
5. **Configure ArcMC Management:** Configure your Event Broker for management by ArcMC, then add it as a host to ArcMC. For instructions, refer to the ArcMC Administrator's Guide.
6. **Set Up Consumers:** Set up one or more consumers to consume event data.

- Deploy at least one of the following: ArcSight Investigate, Logger (6.4 or later), ArcSight ESM (6.11.0 or later), Apache Hadoop, or a third-party consumer.
- Configure consumers to receive events from Event Broker's Kafka cluster.

For more information, see "Consuming Events with Logger" on page 13, "Consuming Event Broker Events with Apache Hadoop" on page 17, or "Consuming Events with Non-ArcSight Applications" on page 16.

Deploying Event Broker

Deploying Event Broker is explained in detail in the ArcSight Installer Deployment Guide, available from the ArcSight documentation repository on the ArcSight software community.

Note: Currently, only deployment of a single master node, with no failover, is supported. A single master node comprises a single point of failure. However, in case of failure, parts of Event Broker functionality will still be available in degraded mode, depending what services were hosted on the master node (for example, the Kafka cluster could still support event data flow through the rest of the Kafka nodes running on the worker node). Multiple master node (multi-master) support is planned for a future release.

Chapter 2: Producing and Consuming Event Data

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

While Event Broker 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 chapter includes the following topics:

- Producing Events with SmartConnectors 11
- Consuming Events with ArcSight Investigate & HPE Vertica 13
- Consuming Events with ESM 13
- Consuming Events with Logger 13
- Consuming Events with Non-ArcSight Applications 16
- Consuming Event Broker Events with Apache Hadoop 17
- Connectors in Event Broker (CEB) 22
- Avoiding Single Points of Failure 22
- Event Broker Sizing 23

Producing Events with SmartConnectors

ArcSight SmartConnectors can publish events to Event Broker Topics. Event Broker supports all SmartConnector types of version 7.6.0 and later.

To publish events you must configure your SmartConnectors to use the Event Broker destination. To send events to multiple topics, you can configure multiple concurrent destinations with the same Event Broker hosts and different topics.

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

Event Broker balances events sent by the SmartConnector between nodes by distributing them evenly between the partitions in the configured topic.

Acknowledgments ensure that Event Broker 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 Event Broker itself has.)

Note: Leader acks 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.

For CEF topics, SmartConnector (version 7.7 or later) encodes its own IP address as meta-data in the Kafka message for consumers that require that information, such as Logger Device Groups.

For more information about SmartConnectors and how to configure a Event Broker destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the ArcSight Product Documentation Community on Protect 724.

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, then push them out to their destination SmartConnectors. You must then configure and enable the Kafka destination on all SmartConnectors.

To upload the Java Keystore files:

1. Prepare the .jks files you want to push and store them in a secure network location.
2. In ArcMC, click **Administration > Respositories > New Repository.**
3. In **Name, Display Name**, and **Item Display Name**, enter KAFKA_JKS
4. Enter other required details as needed, 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.

Consuming Events with ArcSight Investigate & HPE Vertica

Transformed events in the default topic eb-internal-avro, which are in Avro format, can be read by the HPE Vertica database. In turn, once in Vertica storage, event data is accessible for use in ArcSight Investigate searches.

You configure ArcSight Investigate for use with Event Broker as part of the Vertica installer, where you can specify the location of the default Avro topic to which Vertica can subscribe. For instructions, consult the HPE Vertica installer documentation.

Consuming Events with ESM

ArcSight ESM version 6.11.0 or later can subscribe to Event Broker events. ESM requires SmartConnector release 7.6 or later to subscribe to Event Broker topics.

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

Additionally, ESM provides data monitors to monitor Event Broker health.

For instruction on configuring ESM 6.11.0 or later as a consumer, see the ESM Administrator's Guide, available on Protect724.

Consuming Events with Logger

To subscribe to Event Broker topics with Logger, you must configure an Event Broker receiver on Logger 6.4 or later. Logger's Event Broker receivers are consumers for Event Broker's publish-subscribe messaging system. They receive events in Common Event Format (CEF) from Event Broker's topics. An Event Broker receiver connects to ArcSight Event Broker and consumes all events for the topics it subscribes to.

When configuring an Event Broker receiver, specify the consumer group and topic. 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 an Event Broker receiver, refer to the **Configuration > Receivers** section of the ArcSight Logger Administrator's Guide, available for download from the ArcSight Product Documentation Community on Protect 724.

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

Sending Event Broker Data to Logger

For a Logger to be able to consume Event Broker events, the Logger must have an Event Broker receiver configured with the Event Broker hosts, consumer group, and event topic list. SmartConnectors that send data to Event Broker must have an Event Broker 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 ArcSight Logger Smart Message Pool destination on SmartConnectors. The difference is that when the SmartConnectors have an ArcSight Logger Smart Message Pool destination, the event load is balanced by each SmartConnector, but when the SmartConnectors have an Event Broker destination, the event load is balanced by the Loggers.

Additional Loggers can be added to the pool simply by configuring the same Event Broker hosts, consumer group, and event topic List in the new Logger's Event Broker 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 Event Broker destinations to send events to the topic that 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, then configure the Logger's Event Broker receiver to subscribe to this event topic.
- Configure each SmartConnector to publish events to different topics and then configure the Event Broker 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. For example, if there are only five partitions, 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 Event Broker data to Logger (Overview):

1. Configure the SmartConnector:
   - Setup a SmartConnector to publish to a particular Event Topic. Connectors can only send to a single topic for each destination. Additional destinations need configured if each event needs to go to multiple topics.
     Note the number of partitions in the Event Topic.
   - Configure the SmartConnector to have an Event Broker destination. For Logger, use an Event Broker destination.
   - For more information about SmartConnectors and how to configure an Event Broker destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the ArcSight Product Documentation Community on Protect 724.
2. Configure Logger:
   - Create an Event Broker receiver on each Logger in the Logger pool.
   - Configure each receiver to subscribe to the Event 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 Event Broker receiver.
   - Configure each receiver to be in the same Consumer Group.

     For more information on how to configure Logger to receive Kafka events by using Event Broker, see "Consuming Events with Logger" on page 13, and refer to the Receivers section in the Configuration chapter of the Logger Administrator's Guide, available for download from the ArcSight Product Documentation Community.

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 an Event Broker Receiver for 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 all the configuration is set up properly, the Logger pool will subscribe to device type Firewall.

Consuming Events with Non-ArcSight Applications

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

Note: Only use Kafka client libraries that are version 0.11 or later to create consumers.

- All Event Broker nodes, consumers, and producers must be properly configured for DNS/reverse DNS; as well as time, 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.

Using Apache Flume to Transfer Events

One of the applications you could use to transfer Event Broker 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 Event Broker to Apache Hadoop or other storage systems.

Prerequisites

- Event Broker installed: Consult the Event Broker 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 Event Broker as the source agent, your storage system as the sink agent, and ZooKeeper as the channel agent in this file.

To configure Event Broker as the source:

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

Required Kafka Source Configuration

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

To configure the sink:

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

Consuming Event Broker 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 Event Broker events to Hadoop by using Apache Flume.

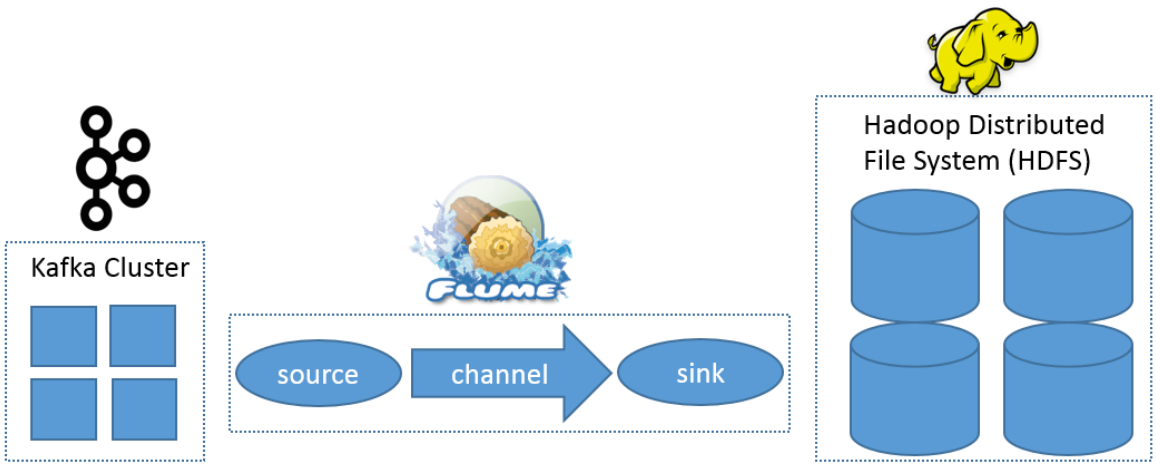
This section describes how to set up the Apache Flume agent to transfer Common Event Format (CEF) events from an Event Broker Kafka cluster to Hadoop Distributed File System (HDFS).

It includes the following topics:

- Architecture for Kafka to Hadoop Data Transfer 18
- Setting Up Flume to Connect with Hadoop 18
- Sample Flume Configuration File 19
- Setting Up Hadoop 21

Architecture for Kafka to Hadoop Data Transfer

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



Setting Up Flume to Connect with Hadoop

About:

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).

Prerequisite:

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" on page 21.

Procedure:

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

   By default, this configuration persists a CEF file every hour. Alternatively, you could choose to roll using events count or file size. If you have high volume of events, HPE ArcSight recommends

using the event count option instead of time, to avoid running out of memory. For more information, refer to Flume HDFS sink in the Flume Users' Guide.

5. Execute the following commands to create the Hadoop cefEvents directory:
   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```
6. Create a configuration file in the Flume conf directory, `bin/flume/conf/`, following the template in "Sample Flume Configuration File" below. In our example we named the file *kafka.conf*. You can name it whatever is appropriate.
   a. Copy `flume-env.sh.template` as `flume-env.sh`.
   b. Edit `flume-env.sh` file and make the following changes:
      - Set JAVA_HOME to the directory where Java was installed on your system.
      - Uncomment the line for JAVA_OPTS:

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

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

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

      ```
      cp <Hadoop install directory>/share/hadoop/hdfs/hadoop-hdfs-2.7.2.jar /<Flume Install directory>/lib
      ```
7. Execute the following command to start Flume from its home directory:
   ```
   bin/flume-ng agent --conf conf/ --conf-file conf/kafka.conf --name tier1 -Dflume.root.logger=INFO,console
   ```
8. After you start Flume, you can find the files on HDFS by running the following command:
   ```
   hadoop fs -ls -R /opt/hadoop/cefEvents
   ```
   This path has to match the HDFS directory path, created in the Hadoop configuration section.

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

Sample Flume Configuration File

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

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

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

```
#Sample Flume/Kafka configuration file
#####################################################
#defines Kafka Source, Channel, and Destination aliases
tier1.sources = source1
tier1.channels = channel1
tier1.sinks = sink1
#Kafka source configuration
tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
tier1.sources.source1.kafka.bootstrap.servers= kafkaIP1:9092, kafkaIP2:9092,…
tier1.sources.source1.kafka.topics = eb-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.
2. Add a hadoop user.
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 Event Broker (CEB)

Event Broker includes support for Connectors in Event Broker (CEB). CEB moves the security event normalization, categorization, and enrichment of connectors processing to the Docker containers environment of Event Broker, while reducing the work done by the system component left outside of Event Broker to collection of raw data (the Collector).

Deploying CEB is performed in ArcMC 2.70 or later, managing Event Broker. For information on managing Event Broker and deploying CEB, see the ArcMC 2.70 Administrator's Guide.

Only syslog connectors are supported with this release.

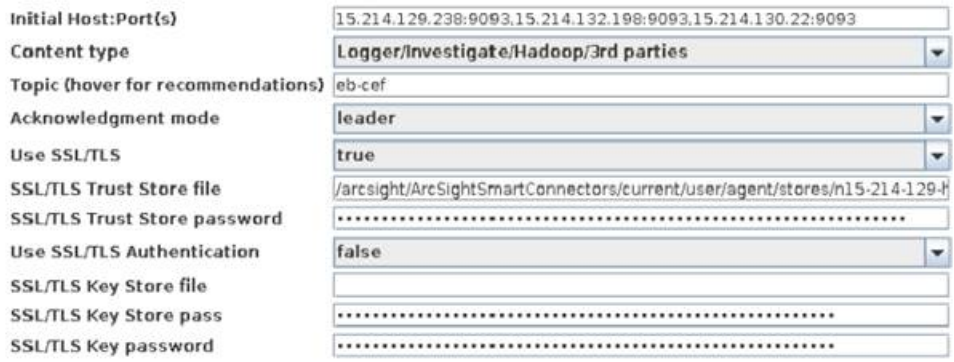
Note: Connectors in Event Broker (CEB) and all related functionality, including Collectors, are provided as **non-production public alpha features.** These features are provided for your testing and evaluation only and should not be considered fully functional, nor are they supported by HPE Support, nor are they guaranteed to be available in the product in the future. Consult the ArcMC Admin Guide, and directions from the ADP product team, for best practices and guidance on how to use these features. **CEB and Collectors should not in any circumstances be used in a production environment.** We welcome questions, comments, and feedback on these features. Please direct any questions or comments to our ADP product team at adp-ceb-alpha@hpe.com.

Avoiding Single Points of Failure

You can configure Event Broker to avoid single points of failure in both the producers sending data to Event Broker (such as connectors), and the consumers subscribing to data from the Event Broker (such as Logger).

For Producers

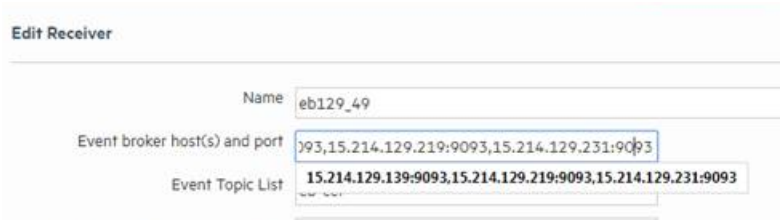
Configure the **Initial Host:Port(s)** parameter field in the Event Broker Destination to include all Kafka cluster nodes as a comma-separated list.



For more information on how Kafka handles this using bootstrap.servers, please see https://kafka.apache.org/documentation/#newconsumerconfigs.

For Consumers

Configure the **Event Broker 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/#producerconfigs.

Event Broker Sizing

The following hardware specifications are intended to server as guidelines for configuring and optimizing your own Event Broker implementation.

Expected Total EPS (Production + Consumption)	Hardware Generation	Disk Type	CPUs/Cores Per CPU	Memory (in GB of RAM)	Network	Minimum # of Nodes
Up to 50k	Virtual or G8+	10-15K RPM SAS or SSD	2/12 (24 cores in total)	32	10Gbit	3
50k-100k	Virtual, Gen8, Gen9	10-15K RPM SAS or SSD	2/12 (24 cores in total)	64	10Gbit	3

Expected Total EPS (Production + Consumption)	Hardware Generation	Disk Type	CPUs/Cores Per CPU	Memory (in GB of RAM)	Network	Minimum # of Nodes
100-250k	Virtual, Gen8, Gen9	10-15K RPM SAS or SSD	2/12 (24 cores in total)	64	10Gbit	5
250k or more	Gen9	10-15K RPM SAS or SSD	2/12 (24 cores in total)	64	10Gbit	5
500k or more	Gen9	10-15K RPM SAS or SSD	2/12 (24 cores in total)	64	10Gbit	7

Notes on the Table

- The above table depicts total EPS (in 1765 byte CEF events); that is, production *and* consumption EPS. For example, if you had 10k EPS inbound, and consumers performing 20k EPS consumption, you would need to size for 30k EPS.
- Hardware is per node, for a minimum 3 node cluster.
- Figures assume no leader acks and no TLS enabled.
  - For leader ACK, include a 30% performance impact.
  - For TLS, consider including a 20-30% performance impact.
- Keep in mind that compression in Kafka is performed on the producer (that is, the Smart Connector) using GZIP. Kafka itself plays no role in compression of data.
- Gen9 hardware is highly recommended. For 24 cores, the DL380 with 2 x 12 core is recommended.
- If using ArcSight Investigate as a consumer, consider the potential performance hit of the Cef2Avro transformation, and allow a 20% increase in CPU utilization. This will generally only have a large impact with very high EPS (250K+).
- Keep hardware the same across your implementation. Don't mix hardware (or physical with virtual machines).
- Adding more nodes is better than installing bigger and faster hardware. Adding more nodes also helps with predicting costs due to new hardware.

Chapter 3: Securing Your Event Broker Deployment

You are responsible for configuring your Event Broker environment securely according to your business needs and requirements. To help you do this, the Event Broker supports Transport Layer Security (TLS) 1.2. Ensure that you have your firewalls configured appropriately for your business needs and deployment.

This chapter includes the following topics:

- Firewall Configuration 25
- Changing Event Broker security mode 25

Firewall Configuration

You can configure your firewall rules to allow access to only the services that are required.

The Event Broker environment requires the following access:

- Kafka uses port 9093, which is TLS-enabled. All customer data is secured by TLS. (If you are using the Vertica database, you must make port 9092 reachable by all Event Broker nodes, consumers, producers, and Vertica nodes, but 9092 is not TLS-enabled.)
- The Event Broker Manager uses port 9999 and 10000 to monitor Kafka. These ports must be mutually reachable between all Event Broker nodes.
- ArcMC communicates with CEB on ports 39001-39010.

By default, ZooKeepers do not use TLS or FIPS to communicate with each other. This communication does not include customer data.

Changing Event Broker security mode

You should decide on a security mode before deployment and setup. In general, the security mode of systems connected to Event Broker (consumers and producers) must be the same as the Event Broker security mode.

TLS is the default configuration. By editing the arcight-installer.properties file, you can enable TLS+CA, as well as FIPS.

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

You can change Event Broker security modes after deployment, but there will downtime for Event Broker and its associated systems, like consumers, producers, and ArcMC. You will need to make sure all Event Broker-associated systems are re-configured as well.

To change security mode (Overview):

1. Stop SmartConnectors from sending events. This will close connections. See the SmartConnector User's Guide for information on stopping SmartConnectors from sending events.
2. Stop all consumers (ArcSight Logger, ArcSight ESM, Vertica Scheduler) from consuming from topics in Event Broker. (There is no need to clear out existing messages from the topics, and the consumers will pick up where they left off later.)
3. If the mode change requires that Event Broker consumer or Event Broker producer restarts, then it must disconnect from Event Broker first. See the consumer or producer documentation. Vertica scheduler does not support different security modes.
4. Undeploy the Event Broker containers.
5. In a text editor, change the arcsight-installer.properties configuration settings.

- predeploy-eb-init-client-auth=false. Set to true to enable TLS+CA.
- predeploy-eb-init-fips=false. Set to true to enable FIPS.

Note: See the Appendix for a complete list of arcsight-installer.properties settings.

6. Redeploy the Event Broker containers.
7. Follow the consumer and producer documentation to reconfigure those applications to align their security modes as Event Broker.
8. Reconnect the consumers and producers. See the respective product documentation for the steps.

Chapter 4: Managing Event Broker

You can manage topic routing and Event Broker infrastructure through ArcMC. Additionally, ArcSight Event Broker provides the Event Broker Manager, a version of Yahoo Kafka Manager, to help you monitor and manage its Kafka services.

For more information about Yahoo Kafka Manager, refer to https://github.com/yahoo/kafka-manager.

For more information about Kafka monitoring, refer to the monitoring section of the Apache Kafka documentation.

This chapter includes the following topics:

- Managing Event Broker through ArcMC 27
- About the Event Broker Manager 27
- Command Reference 37
- Graceful Server Reboot 38

Managing Event Broker through ArcMC

You can create topics and routing rules, monitor Event Broker metrics, and receive notifications about Event Broker status through ArcSight Management Center (ArcMC).

Monitored Event Broker parameters include CPU usage, memory, disk usage, throughput, EPS (Events per Second) In, event parsing errors, stream processing EPS, and stream processing lag.

Enabling Event Broker management through ArcMC

To enable Event Broker management in ArcMC, add your Event Broker as a host to ArcMC. The procedure for adding Event Broker as a host is explained in detail in the ArcMC Administrator's Guide, available from the ArcSight software community.

The ArcMC Administrator's Guide also explains in detail how to view the list of Event Broker consumers, manage topics, routing rules, monitored metrics, and enabling notifications.

About the Event Broker Manager

The Event Broker Manager enables you to manage your clusters, topics, and partitions. It enables the following monitoring and management options:

- 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 Event Broker Manager

Only users that can log into the Event Broker server can access the Event Broker Manager. These users can access the Event Broker Manager by using their local web browser directly from any of the Event Broker nodes or by using SSH forwarding from the Kubernetes worker node where Kafka is running.

You can connect to the Event Broker Manager with most browsers, including Chrome, Firefox and Internet Explorer. For a list of browsers supported in this release, refer to the ADP Support Matrix, available for download from the ArcSight Product Documentation Community on Protect 724.

To access Event Broker Manager:

1. On an Event Broker node, run the command `kubectl get service -n arcsightventbroker1` to get the list of services.
2. Locate the Event Broker Manager service eb-kafkamgr-svc and note its IP and port number.

To connect directly from an Event Broker server node:

1. Log into the Event Broker server.
2. With a supported browser, connect by using the IP and port of the Event Broker manager (as shown previously).

   `http://127.0.0.1:<Port>`

Once you connect, the browser displays the Clusters page. See "Managing Clusters" on the next page.

To connect from your local machine:

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

   `ssh -L <Event Broker port>:<Event Broker IP:port> <master_node_IP>`
2. With a supported browser, connect by using the following URL:

   `http://127.0.0.1:<Port>`

Once you connect, the browser displays the Clusters page. See "Managing Clusters" on the next page.

Managing Clusters

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

Location: Clusters

Click the *Cluster Name* link. The Event Broker Manager displays the Cluster Summary page. See"Viewing Information About a Cluster" below.

To edit the cluster:

1. Click **Modify**. The Event Broker 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:

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. See "Managing Topics" on page 31.

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. See "Managing Brokers" below.

Managing Brokers

On the **Brokers** page, you can view overview information on all of your broker nodes and drill down into a broker for more information.

> **Note:** The term *Brokers* is used synonymously with *Event Broker Nodes* in ArcMC. Both terms describe a single node running Kafka.

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. See "Viewing Broker Details" below.

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.

From here you can view the following:

- "Summary" on the next page
- "Metrics" on the next page
- "Messages count" on the next page
- "Per Topic Detail" on the next page

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" on page 33.

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: These topics are installed by default:

- __consumer_offsets
- _schemas
- eb-internal-datastore
- eb-internal-stream-processor-metrics.

These are used internally by Event Broker and should not be modified.

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" on the next page.

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. 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

> **Note:** You cannot delete topics once they have been created.

To open the Add Topic page:

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

To create a new topic:

Fill in the fields and click **Create**. For a discussion of field values, consult the Apache 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.

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

Viewing Topic Details

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

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

To view information on a specific topic:

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

From here you can view the following:

- "Topic Summary" below
- "Metrics" below
- "Operations" below
- "Partitions by Broker" on the next page
- "Consumers consuming from this topic" on page 35
- "Partition Information" on page 35

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 reassign partitions, generate partition assignments, add partitions, update the topic configuration, and manually assign topics to broker nodes.

To reassign partitions:

Click **Reassign Partitions**.

To generate partition assignments:

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

To add a partition:

1. Click **Add Partitions**.
2. Enter the new number of partitions.
3. Select the topics and broker nodes.
4. Click **Add Partitions**.

To update the 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 Kafka Manager, users will see different offset values between Binary (ESM) and CEF (Investigate or Logger) topics. In CEF topics, the offset value can generally be associated with number of events that passed through the topic. Each message is an individual event. However, that same association cannot be made in Binary topics.

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" on page 30.

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.

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.

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.

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**.

Command Reference

The following Kubernetes commands are commonly used for the operation and administration of Event Broker.

Description	Command
Find the nodes in the cluster	`# kubectl get nodes`
List the nodes with a specific label applied, for example "kafka"	`# kubectl get nodes -L kafka#`
Find status of the Event Broker application modules	`# kubectl get pods -n arcsighteventbroker1 -o wide`
Find the pod name for a specific module, like the Event Broker Manager (Kafka Manager)	`# kubectl get pods -n arcsighteventbroker1 | grep kafka-manager`
Restart a specific module, like the Event Broker Manager (Kafka Manager)	`# kubectl get pods -n arcsighteventbroker1 | grep kafka-manager`

Description	Command
Delete a specific module, like the Event Broker Manager (Kafka Manager)	`# kubectl delete pod eb-kafka-manager-707541454-cmxhs -n arcsighteventbroker1`

Get logs from a specific module, like the Schema Registry	`# kubectl  get pods -n arcsighteventbroker1 | grep schemaregistry`
	`eb-schemaregistry-2567039683-9l9jx   1/1       Running   1          18d`
	`# kubectl logs eb-schemaregistry-2567039683-9l9jx -n arcsighteventbroker1 | more`

Execute a command inside one of the application pods	`# kubectl get pods -n arcsighteventbroker1 | grep c2av`
Run a single command	`# kubectl exec eb-c2av-processor-0 -n arcsighteventbroker1 -- cat`

Open a bash shell first, to run multiple commands inside	`# kubectl exec eb-c2av-processor-0 -n arcsighteventbroker1  -it bash`
	`# cat /eb/sp/config/stream.properties | grep num.stream.threads`
	`num.stream.threads=6`
	`#`

Graceful Server Reboot

You may need to intentionally reboot your Event Broker as part of planned hardware or system maintenance. A graceful reboot consists of deliberately shutting down the cluster and then restarting it.

Server Console Access

If you have access to the server console and can restart (not just reboot) each server, follow this procedure to begin a graceful reboot:

1. If possible, turn off all the data producers sending events to Kafka (such as connectors or any third-party producers) and wait for Kafka to process all incoming events.
2. If possible, turn off all the data consumers like (such as Logger or any third-party consumers).
3. Shut down each of the Kubernetes worker node servers.
4. After all worker nodes are shut down, shut down the Kubernetes master node server.

Now start up the cluster as follows:

1. Bring up the Kubernetes master server.
2. After the master server is up, bring each of the Kubernetes worker node servers.
3. Once all the Event Broker pods are up and in the running state (checked by `kubectl get pods`), now re-start the producers and consumers (if you have stopped them before shutdown).

After a server reboot, all Kubernetes and Docker services will start automatically.

SSH Access Only

If you only have SSH access to the servers and can only reboot them, follow this procedure for a graceful reboot:

1. Undeploy Event Broker from the ArcSight Installer UI page
2. Use `kubectl drain` to to remove each worker node from service
3. Use `kubectl get nodes` to get the worker nodes
4. Remove all pods from each node each node using `kubectl drain <node name>`

Note: A warning may be displayed about not able to evict pods. This message may be ignored.

Now reboot the cluster as follows:

1. Reboot the master node and wait for the reboot to finish.
2. Reboot the worker nodes and wait for all of them to finish their reboots.
3. Use `kubectl uncordon` to bring each of the worker node back in service
4. Deploy Event Broker from the ArcSight Installer UI page

Chapter 5: Managing Event Broker Topics

You can manage your Event Broker topics through Event Broker Manager or through ArcMC.

This chapter includes the following topics:

- Default topics 40
- Topic Configuration 40
- Data redundancy and topic replication 41
- Managing topics through ArcMC 41

Default topics

Event Broker is deployed with several default topics. You can use one of these or configure your own. Topic names are case-sensitive.

Default Topic Name	Description
eb-esm	Binary security event. Supports ESM as a consumer. Use for all ESM events. You can add other topics for ESM events, but cannot create routes for them.
eb-cef	Use for CEF 0.1 or CEF 1.0 events. You can create routes to filter events in this topic.
eb-con-syslog	Default topic for CEB.
eb-internal-stream-processor-metrics	This topic is for internal use only. Do not configure your SmartConnector destinations to send events to this topic.
eb-internal-avro	This topic is for internal use only. Do not configure your SmartConnector destinations to send events to this topic.
__consumer_offsets	This topic is for internal use only. Do not configure your SmartConnector destinations to send events to this topic.
_schemas	This topic is for internal use only. Do not configure your SmartConnector destinations to send events to this topic.
eb-internal-datastore	This topic is for internal use only. Do not configure your SmartConnector destinations to send events to this topic.

Topic Configuration

HPE ArcSight recommends using different topics for categorization, for example, firewall events in one topic and anti-virus events in another.

- Configure topics based on data isolation requirements or categorization. If you route only to categorized topics, then events are not sent to Vertica for use by ArcSight Investigate.
- Configure partition count for your topics based on throughput and number of consumers. The partition count should be at least equal to the total number of present (and future) consumers.
- Configure the replication factor based on how important the events are. The recommended replication factor for new topics is 2. While you can replicate every topic to every node in the cluster, this is not recommended because the extra traffic reduces throughput and increases disk space requirements.

Data redundancy and topic replication

When setting up Event Broker, you can specify the number of copies (replicas) of each topic Event Broker should distribute.

Kafka automatically distributes each event in a topic to the number of broker nodes indicated by the topic replication level specified during Event Broker configuration. While replication does decrease throughput slightly, HPE 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.

A topic replication level of 1 means that only one broker will receive that event. If that broker goes down, that event data will be lost. However, a replication level of 2 means that two broker nodes will receive that same event. If one goes down, the event data would still be present on the other, and would be restored to the first broker node when it came back up. Both broker nodes would have to go down simultaneously for the data to be lost. A topic replication level of 3 means that three broker nodes will receive that event. All three would have to go down simultaneously for the event data to be lost.

When you add new consumers, you don't need to update your producers. Event Broker handles the distribution and replication for you.

Refer to the Apache Kafka documentation for more information.

Managing topics through ArcMC

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

A *route* is a rule that directs Event Broker to duplicate events that meets certain criteria from a source topic to the route's destination topic. Rules are defined using event field names and expected values.

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

Note: Only CEF text format events may be routed. Binary security events in the eb-esm topic may not be routed.

Chapter 6: Licensing

The Event Broker licensing check process uses the ArcMC PD file and ADP ArcMC licenses for EB license check during Kafka pod startup.

Applying a New License File

Event Broker installs with a default 30-day instant-on evaluation license, issued in the form of an XML file. To extend Event Broker functionality past 30 days, you must apply a new ArcMC ADP license file to each worker node.

Only the ArcMC ADP license is supported by Event Broker. Other ADP licenses, such as ADP Logger, and other ArcMC licenses, such as ArcMC CHA, are not supported.

Note: It is **strongly** recommended that you do not use the 30-day evaluation license for a production Event Broker. Plan and apply a proper license before the expiration date, to ensure continuity of functionality and event flow.

Because applying a license requires a restart, plan your license update time to minimize Kafka down time.

To apply a new license file:

1. On each worker node where Kafka is running (`eb-kakfa-0` to `eb-kakfa-n`) rename the `ArcSight ArcMC ADP 2.7.dat` file to `license.xml` and upload it to the `/opt/arcsight/k8s-hostpath-volume/eb/autopass/` directory.
2. Redeploy Event Broker for the new license to take effect.

To confirm the license has been applied, run the following command on each node, and confirm the property `date.expiration=XXXX/XX/XX` in the output.

```
kubectl logs eb-kafka-0 -n arcsighteventbroker1 | more
```

How to check the license log

For each Kafka pod (eb-kakfa-0 to eb-kakfa-n) check as follows:

```
# kubectl logs eb-kafka-0 -n arcsighteventbroker1 | more

....

ArcSight Data Platform Event Broker License Check
```

```
WARN: license file not found

Sep 30, 2017 7:10:43 PM java.util.prefs.FileSystemPreferences$1 run

INFO: Created user preferences directory.

Autopass license is valid

date.expiration=2017/10/30

software.model=ArcMC Software

arcmc.feature.adp.managed=false

arcmc.limit.daily.data=0GB

component.name=arcmc

license.trial=true

arcmc.enabled=true

eb.license.enabled=true

...

#
```

License Check Behavior

The following table shows the behavior of the license check in a variety of scenarios.

Scenario	EB License Status	Expected EB Behavior	Kafka pod startup log
No license file, EB running after initial install	30 day instant on license from day of install for EB	Kafka pods will be up and running without issues.	eb.license.enabled=true
No valid license file, EB running for 30 days	No valid license for EB	When a Kafka pod is restarted after 30 days, it will exit without coming up.	eb.license.enabled=false
Valid ArcMC ADP license (permanent license, no expiration)	Valid EB license forever	When Kafka pods are initially deployed or restarted later, they should be up and running without issues.	eb.license.enabled=true

Scenario	EB License Status	Expected EB Behavior	Kafka pod startup log
Expired ArcMC ADP license (with expiration)	Valid EB license till the date of expiry.	When Kafka pod is restarted after the date of expiry, it will exit without coming up.	eb.license.enabled=false
Valid ArcMC CHA license	Not a valid license for EB	When Kafka pod is restarted or on first deployment, it will exit without coming up.	eb.license.enabled=false
Valid ArcMC nonADP license	Not a valid license for EB	When Kafka pod is restarted or on first deployment, it will exit without coming up.	eb.license.enabled=false

Chapter 7: Troubleshooting

These troubleshooting tips may prove helpful in resolving issues with Event Broker.

This chapter includes the following topics:

- Diagnostic Data and Tools 47
- Verifying the health of the Event Broker cluster 48
- Master or Worker Nodes Down 49
- Diagnosing Common Event Broker Issues 50
- Tuning Event Broker Performance 54

Diagnostic Data and Tools

Event Broker includes a diagnostic script (`eb-diag.sh`) for the collection of diagnostic data. Diag.sh is found in the web service container.

To run Event Broker diagnostic tools:

1. On the master server, find the Docker container ID of the web service container. (In this example 278e86760803)

```
$ docker ps | grep atlas_web-service

278e86760803        localhost:5000/arcsightsecurity/atlas_web-
service@sha256:c25b023afa7b7054de6aa188ed2802d24312f3c5de87b6537aa3e93747
6376d8   "/bin/bash -c 'source"

$
```

2. Copy the script archive from web service container (In this example 278e86760803)

```
$ docker cp 278e86760803:/eb/ws/eb_diag/eb_diag.tgz .

$ tar -tzf eb_diag.tgz

vertica-diag.sh

eb-diag.sh

eb-sys-diag.sh

$
```

3. Extract the diagnostic script and run it

```
$ tar -xvf eb_diag.tgz eb-diag.sh
```

```
$ sh eb-diag.sh
```

Verifying the health of the Event Broker cluster

Verify the health of each container: run `kubectl get pods --all-namespaces -o wide` to list pods and their status.

View Kubernetes logs for each container: run `kubectl logs`

```
# kubectl logs -n arcsighteventbroker1 [POD ID/NAME]
```

```
# kubectl logs -n arcsighteventbroker1 [WEB SERVICE POD ID/NAME] -c atlas-web-service
```

Verify data flows through the system: check any of the following.

- In ArcMC, review the EPS graph. This indicates whether events are flowing through the stream processor (routing and transforming).
- In Vertica server, check the Kafka scheduler status to see event count and reject count. You should be able to see the event count increasing.

```
# ./install-vertica/kafka_scheduler status
```

- Check the Kafka manager offset with the select count(*) in Vertica. The count should increase over time. (for example, SELECT COUNT (*) FROM investigation.events;)
- All topics: check the offset for each topic in Event Broker Manager. you should see the value increasing.

Verify that Web Service APIs are healthy:

- Check logs of the web service container (see command above)
- Make sure the port is bound:

```
# netstat -lntp | grep 38080
```

- Verify Vertica Scheduler is running.
- Check the Kakfa Scheduler status.

```
# watch ./root/install-vertica/kafka_scheduler  status
```

- Check whether the offset is increasing in the status output. If not, then there may no data in the Avro topic, or if Avro contains data there may be a problem.
- Verify the topic partition count and distribution.

- Check that the configured partition count matches its expected value
- Check the partition count or replication factor for the topic using Event Broker Manager.

Master or Worker Nodes Down

This section describes what behavior can be expected if the master node or one or more worker nodes goes down.

- Kubernetes worker nodes will continue running even when master is down, but if they reboot then they will not be able to find the master and will fail.
- All services running on the master node will become unavailable.
- Event Broker Web Service running on the master node becomes unavailable.
  - The services (Routing Stream Process) and integration (ArcMC management) that depend on the Web Service will fail.
  - Any other Event Broker service (Transform Stream Process, Schema Registry, Kafka Manager) that was running on the master will get scheduled by Kubernetes on other worker nodes depending on system resources available.
  - If the master node was labeled for Kafka and/or ZooKeeper deployment,then those instances will fail but the cluster will still work with rest of the instances on worker nodes
- The NFS server, which runs on the master node, will become unavailable.
  - Kafka and ZooKeeper do not depend on NFS storage and use local Kubernetes worker node storage. They would will be available for event processing with some limitation.
  - The beta feature Connector in EB (CEB) will be affected, since it depends on NFS storage, which is configured on master server.
- DNS service (`kube-dns`) runs on the master server will become unavailable
  - Worker nodes would lose ability to resolve hostnames, except for those that had already been resolved, and which may be cached for some period.
- Any of the Event Broker service instances that are not tied to a worker node (such as Transform Stream Process, Routing Stream Process, Schema Registry, or Kafka Manager) running on the downed worker node will be scheduled by Kubernetes on other worker nodes, depending on system resources available on other worker nodes.
- Depending on system resources on other worker nodes, Event Broker service instances that use labels like Kafka and ZooKeeper will be automatically scheduled by Kubernetes on other worker nodes (if there were additional worker nodes that were already labeled).
- Likewise, c2av processing may cease if the worker node containing the `eb-c2av-processor` processor goes down and system resources prevent Kubernetes from automatically rescheduling processing on another worker node.

- If automatic re-scheduling of service instances does not occur (such as Kakfa, Zookeeper, or c2av processing), use the following manual command to delete all service instances from the failed node and move the services to other nodes:

```
# kubectl delete node <Failed_Node_IP>
```

Diagnosing Common Event Broker Issues

The following can help to diagnose common Event Broker issues.

Potential DNS Resolution Issue

The Event Broker application pods that depend on hostname resolution from DNS could fail. For example, the Schema Registry pod will be in a crash loop status, with following error message in the Schema Registry logs:

```
# kubectl logs eb-schemaregistry-1138097507-1jxbn -n arcsighteventbroker

...

org.apache.kafka.common.config.ConfigException: No resolvable bootstrap
urls given in bootstrap.servers

...
```

The following steps will be useful in debugging this DNS resolution issue. The key is that the bootstrap host name given should be resolvable from within the pod, which can be verified as follows. Find the schema registry pod name:

```
# kubectl  get pods -n arcsighteventbroker1 | grep schemaregistry

eb-schemaregistry-2567039683-9l9jx   1/1       Running   1          18d
```

Find the configured bootstrap server

```
# kubectl logs eb-schemaregistry-2567039683-9l9jx -n arcsighteventbroker1
| grep "bootstrap.servers ="

bootstrap.servers = [n15-214-137-h51.arst.usa.hp.com:9092]
```

Use `ping` to check if the hostname is DNS resolvable. If it is resolvable, you will see a successful ping.

```
# kubectl exec  eb-schemaregistry-2567039683-9l9jx -n
arcsighteventbroker1  -- ping -c 1 n15-214-137-h51.arst.usa.hp.com | grep
transmitted
```

```
1 packets transmitted, 1 packets received, 0% packet loss
```

If not, you will see an error

```
# kubectl exec  eb-schemaregistry-2567039683-9l9jx -n
arcsighteventbroker1  -- ping -c 1 bad.dns.hostname.arst.usa.hp.com |
grep transmitted

ping: unknown host
```

If hostname is not resolvable, please check the DNS configuration on the system.

Event Broker Cluster Down

The number of nodes required to keep an Event Broker cluster operating depends on the replication factor. If the replication factor is only 1, which is not recommended, then all Kafka nodes in the EB cluster need to be up to make the EB cluster function correctly. In general, if the replication factor is N, then the system will tolerate up to N-1 server failures without losing any records committed to the log.

Pod Start Order

After deployment, pods are configured to start in the following order (downstream pods will not start until the dependencies are met.)

1. A quorum of ZooKeeper pods in the cluster must be up (2 of 3, or 3 of 5). The total number of ZooKeepers must be odd.
2. All Kafka pods must be up
3. Schema Registry pod must be up
4. Bootstrap Web Service, Event Broker Manager
5. Transformation Stream Processor, Routing Stream Processor

Cannot query ZooKeeper

This can occur when running the `kubectl get pods` command to get status of the pods, downstream pods (as defined in the pod start order) do not stay up, and status is a 'CrashLoop'-type error.

- Check that ZooKeeper pods are running.

- If the ZooKeeper pod status is `Pending`, you may not have labeled the nodes correctly (zk=yes). Verify that the nodes are labeled using the `kubectl get nodes -L=zk` command.
- Verify that you configured an odd number of ZooKeepers in the arcsight-installer.properties eb-zookeeper-count attribute.
- Check the ZooKeeper pod logs for errors using the `kubectl logs <pod name> -n arcsighteventbroker1.`

Common Errors and Warnings in ZooKeeper logs

- **Quorum Exceptions:** A leader cannot be elected. If you see this type of error, check the conditions above.
- **Socket error:** this can occur if there are too many connections. The solution is to restart the pod using the `kubectl delete pod <pod_name> -n arcsighteventbroker1.` The pod will be recreated automatically.

Common Errors and Warnings in Kafka logs

Cannot Register ID: In some cases, a broker node cannot register its ID. This can be caused by multiple broker nodes with the same ID. This is a rare situation that can occur when you are adding and removing nodes from the cluster and you do not define the cluster properly. Connect to each system running a Kafka broker and check the assigned broker.id value of each, in `/opt/arcsight/k8s-hostpath-volume/eb/kafka/meta.properties.` The broker.id value defined on each Kafka node must be unique.

SSL Connection Errors: These are warnings that occur if there is a connection issue between Kafka and consumer or producer.

Cannot communicate with other brokers: Host names may not be configured properly. It is possible that the node cannot perform reverse lookup or that DNS is not set up properly.

Event Broker default topics not created on first deployment: In this instance, the Bootstrap Web Service log contains 500 response code (the response from the Schema Registry), and topics are not created. Try undeploying the Event Broker containers, and then redeploy them.

One or more connectors cannot send data to Kafka: Check the following:

- The connection configuration is set properly in the connector.
- The encryption mode (TLS, TLS+FIPS, TLS+CA, TLS+FIPS+CA) is the same for both the Connector and Event Broker.
- Make sure you can connect to the Kafka port on the system and that there are no network issues.

Cannot retrieve the certificate error when connecting: Make sure that time is synced across all systems in the data pipeline.

• Check whether the Kafka pod is down. Did you configure the connector with only one broker address and that broker is down; If you expect that there are multiple brokers, they must be all configured in connector as a comma-separated list;

• If the replication factor is set to 1 and a Kafka broker is down, data will not be sent through Event Broker. Fix the broker issue to bring it back up. In general, topics should be configured with replication factor greater than 1 so as to prevent this scenario.

Kafka is resyncing: This may cause event throughput slowdown, but will not stop event flow.

Vertica cannot read events from Kafka: After verifying that the Event Broker is still up, check the following:

• For a new setup: Check that Kafka scheduler is configured to communicate to Kafka port 9092. Also, check the network connection.

• For an existing setup (with Vertica consumers): Offset may not be recognized: In this scenario, the Kafka scheduler fails to recognize offset IDs of messages that are in the topic. It can happen if the Kafka scheduler unexpectedly stops reading from the topic, and then is restarted.

Solution: Execute the kafka_scheduler delete command to delete the metadata. After doing this, immediately run the kafka_scheduler create command to set up the scheduler.

• Existing set up: You have configured all brokers that contain the topic the consumer connects to, and the brokers which are configured for that consumer are down.

An EB component crashes: Check the following:

- Check the container start up order (above). Have any of the dependency pods not started or crashed?
- It could be that the JVMs require more memory than the system has available.
- Check the number of open sockets.

Event Broker EPS is lower than expected: Check resource constaints on Event Broker nodes, such as CPU, memory, or disk space. Also, check usage with ArcMC.

Network bottleneck: In this case, the Stream Processor is not able to keep up with transformation, or is resource-constrained in some way. In ArcMC, the Stream Processor metric will be lower than the connector EPS. Check that you have sufficient resources, memory, CPU.

Continuous network failures: This may be related to the management of TCP/IP resources. TIME_WAIT is the parameter which indicates the amount of time the node will try take to finish closing a connection and the amount of time before it will kill a stale connection. Try reducing the value from its default. Edit the file `/etc/sysctl.conf` and add these lines to the end of it (or edit the existing values):

Decrease TIME_WAIT seconds

```
net.ipv4.tcp_fin_timeout = 10
```

Recycle and Reuse TIME_WAIT sockets more quickly

```
net.ipv4.tcp_tw_recycle = 1
```

```
net.ipv4.tcp_tw_reuse = 1
```

After editing the file you should run

```
$ sysctl --system
```

Tuning Event Broker Performance

The following can help improve the performance of Event Broker.

Increasing Stream Processor EPS

You can increase Stream Processor EPS by adding more stream processor instances using the ArcSight installer configuration UI. The configuration in the ArcSight installer configuration UI affects transforming stream processor only (c2av). It does not change the routing stream processor. You cannot modify the number of streams in the routing stream processor.

When you change this value, you do not need to redeploy Event Broker. Please note that this change will increase the number of pods. You will see this difference when you run the `kubectl get pods --all-namespaces` command.

Increasing Kafka retention size/time

You can change the value of retention size or time in any topic using Event Broker Manager after deploying Event Broker containers, and it will be applied immediately. You can change this while events are flowing through the topic.

To change the default values *before* you deploy, change the values in the arcsight-installer.properties file.

Adding a new worker node

To add a new worker node, label the new node (delete or overwrite existing label with a different label). Remove the label from the old node. Kubernetes should start Kafka on the new node.

Then, reassign partitions on the new node. Data copying will take some time to complete.

Glossary

A

Apache Avro
A data serialization system. Avro enables highly space-efficient event storage.

Apache Flume
A service for efficiently collecting, aggregating, and moving large amounts of log data.

Apache Hadoop
A software framework that enables the distributed processing of large data sets across clusters of computers. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. You can configure Hadoop as an Event Broker consumer.

Apache Kafka
An open source distributed publish-subscribe messaging system installed as part of Event Broker.

Apache ZooKeeper
A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. ZooKeeper's architecture supports high availability through redundant services. ZooKeeper is installed as part of Event Broker, which uses it to coordinate the Kafka cluster.

ArcMC
ArcSight product that centrally manages other ArcSight products, including Event Broker, Logger, and connectors.

arcsight-installer.properties
Properties file that controls many Event Broker settings.

B

broker
An instance of the Kafka server software.

C

CEF
Common Event Format (CEF) is an extensible, text-based, high-performance format designed to support multiple device types in the simplest manner possible. Various message syntaxes are reduced to one-matching ArcSight Enterprise Security Manager (ESM) normalization. Specifically, CEF defines a syntax for log records comprised of a standard header and a variable extension, formatted as key-value pairs. This format contains the most relevant event information, making it easy for event consumers to parse and use them.

channel
In Apache Flume, a buffer that stores events, until a sink has successfully written the them.

cluster
A collection of brokers working together to increase throughput and durability.

consumer
A process that subscribes to one or more topics and processes the feed of messages.

consumer group
A logical grouping of several consumers, where only one consumer in the group will process each message.

consumer offset
The read position for a consumer in a partition.

D

device group
In Logger, a category of named source IP addresses called devices. Device groups can be associated with storage rules that define the storage group where events from specific devices are stored. Refer to your Logger documentation for complete details.

Docker
A software technology providing operating-system-level virtualization also known as containers, promoted by the company Docker, Inc.

E

ESM
Enterprise Service Management (ESM) is an ArcSight product that...

Event Broker Manager
Administration tool packaged with Event Broker. Equivalent to Yahoo Kakfa Manager.

F

FIPS
Federal Information Processing Standards (FIPS) are standards developed by the U.S government for use in computer systems by non-military government agencies and government contractors. Specifically, FIPS PUB 140-2, is a U.S. government computer security standard used to approve cryptographic modules.

H

HDFS
Hadoop Distributed File System (HDFS) is a Java-based file system that provides scalable and reliable data storage, and was designed to span large clusters of commodity servers.

I

Investigate
Investigate is an ArcSight product that executes fast data searches and includes advanced analytics.

K

k8s
Abbreviation for Kubernetes.

Kubernetes
An open-source system for automating deployment, scaling and management of containerized applications. It aims to provide a platform for automating deployment, scaling, and operations of application containers across clusters of hosts. It supports a range of container tools, including Docker.

L

leader
The broker containing the original replica of a partition, and manages that data.

Logger
An ArcSight product that receives event data and stores it for retrieval and analysis. You can configure Logger as an Event Broker consumer. Refer the Logger Administrator's Guide for complete details.

N

node
The machine a Kafka instance is running on.

O

offset
A sequential number identifying the location of a message in a partition. The sum of partition offsets is the total number of events in the topic.

P

partition
A segment of a topic. There can be one or more partitions for each topic. The number of partitions limits the maximum number of consumers in a consumer group.

pool
A logical grouping of Loggers. Loggers in a pool belong to the same consumer group and subscribe to the same topics.

producer
A process that publishes messages to a topic. In Event Broker, this is an ArcSight SmartConnector.

publish
The action of sending topics to the Event Broker. A producer publishes event on a given topic.

Q

quorum
The set of all in-sync replicas for a particular partition. Replicas are considered in-sync if they are caught-up to the leader. The leader waits until a majority of replicas have received the data before considering it to be committed. On leader failure, a new leader is elected through the coordination of a majority of the followers. If there is an odd number of replicas, a majority is ensured. Any replica in the quorum can become the leader. This enables the producer to continue to publish messages and the consumer continues to receive the correct messages, even when there is failure.

R

receiver
In Logger, the process that receives events, captures event data, and populate each event with information about its origin. Refer to the Logger Administrator's Guide for complete details.

replica
A copy of a partition. There can be one or more per partition; even if there is no redundancy, the original is still called a replica.

replication factor
The number of times a topic is duplicated across Kafka nodes. A replication factor of 3 means that the topic is copied to 3 Kafka nodes.

route
A rule that directs Event Broker to copy events that meets certain criteria to the route's destination topic. A route leaves a copy of the event in both the source and destination topics.

S

scheduler
Manages and tracks the job process for Kafka.

sink
In Apache Flume, the sink forwards events to the storage destination.

SmartConnector
An ArcSight product that collects event data from objects on your network. They normalize the data in two ways: normalizing values (such as severity, priority, and time zone) into a common format, and normalizing the data structure into a common schema. You can configure SmartConnectors as Event Broker producers. Refer to the SmartConnector User's Guide for complete details.

source
In Apache Flume, a source sends events to Flume.

Stream Processor
The Event Broker mechanism for processing incoming events and converting their data format from CEF to AVRO. Also known as c2av.

subscribe
The action a consumer takes in order to receive the events that are published to a topic. A subscriber can receive the events published while the subscriber is active, or it can request events "from the beginning of time" the first time its consumer group is seen. From then on it will retrieve events since the last time a consumer for its group retrieved events.

T

TLS
Transport Layer Security. Enabled in Event Broker by default.

topic
A feed of messages relating to the same category.

transform
To convert data from one format to another. For example, Event Broker transforms CEF events into Avro format, where they can be stored in Vertica.

V

Vertica
HPE Vertica is an advanced SQL database analytics portfolio that enables you to run SQL on Hadoop, and leverage scalable predictive analytics and a comprehensive library of built-in analytical functions.

Y

Yahoo Kafka Manager
An open source tool for managing Apache Kafka. Event Broker includes a version of Yahoo Kafka Manager

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 system, click the link above and an email window opens with the following information in the subject line:

Feedback on Administrator's Guide (Event Broker 2.11)

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 arc-doc@hpe.com.

We appreciate your feedback!