Hewlett Packard
Enterprise

HPE Security ArcSight ADP Event Broker

Software Version: 2.02

Deployment Guide

July 24, 2017

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 Development LP shall not be liable for technical or editorial 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. Hewlett Packard Enterprise Development LP 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 Development LP 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

Support

Contact Information

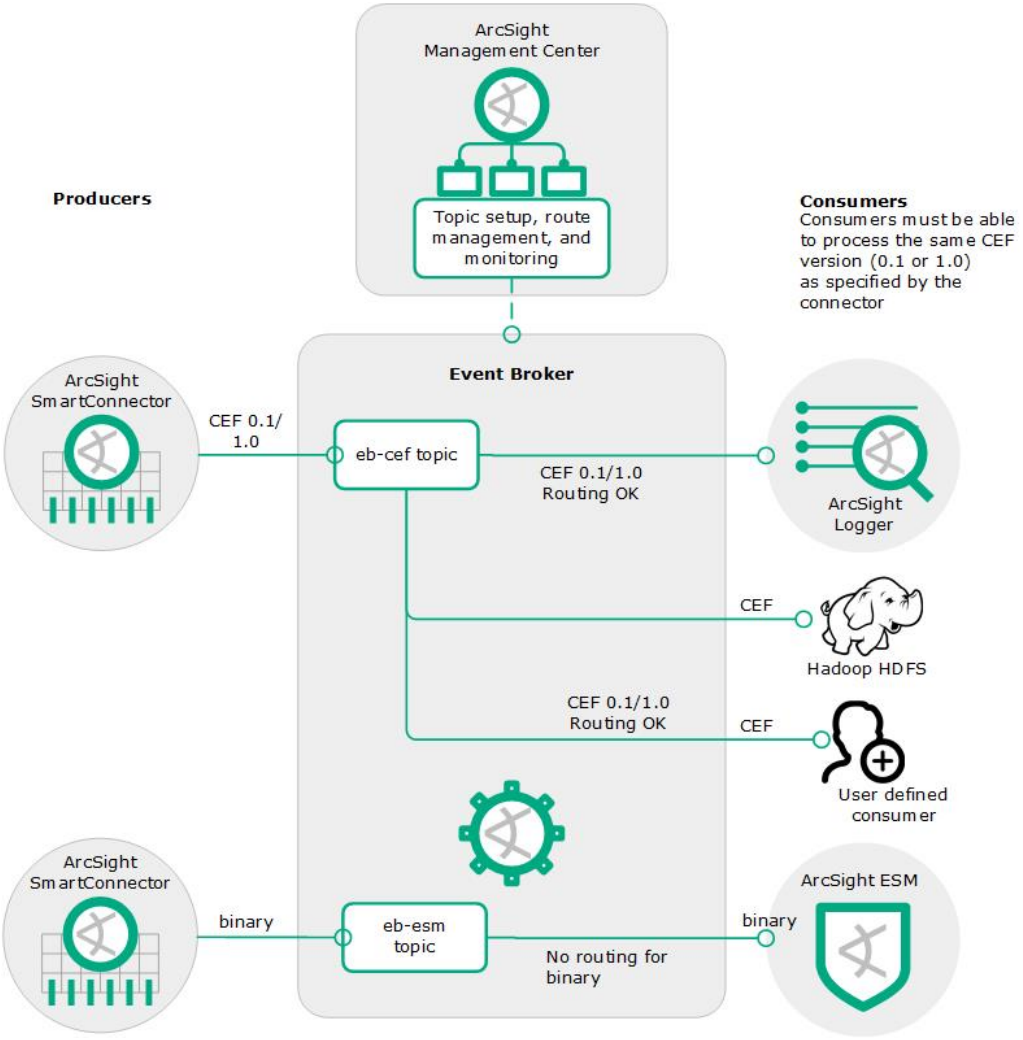
Phone	A list of phone numbers is available on the HPE Security ArcSight Technical Support Page: https://softwaresupport.hpe.com/documents/10180/14684/esp-support-contact-list
Support Web Site	https://softwaresupport.hpe.com
Protect 724 Community	https://www.protect724.hpe.com

Contents

About ArcSight Event Broker

ArcSight Event Broker centralizes event processing and enables topic sorting and event routing, which 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 Kafka broker or cluster of broker nodes, you can use ADP SmartConnectors to publish data, and subscribe to that data with ADP Logger, ArcSight Investigate, Apache Hadoop, or your own consumer.



Component	Description
ArcSight Installer	A web application for deploying and configuring the ArcSight components, including ArcSight Investigate and Event Broker. The components are managed in a Kubernetes cluster.
ArcSight SmartConnectors	SmartConnectors collect and normalize event data from devices on your network. Connectors normalize event 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. SmartConnectors can then filter and aggregate events to reduce the volume of events sent to the system. ArcSight SmartConnectors, installed and maintained separately, are producers that publish data to Event Broker. You can subscribe to data managed by Event Broker with ArcSight Investigate, ArcSight ESM, ADP Logger, Apache HDFS, or your own third-party consumer.
Event Broker	ArcSight Event Broker centralizes event processing, enabling you to take advantage of scalable, high-throughput, multi-broker clusters for publishing and subscribing to event data. Event Broker coordinates and manages data streams, which enables your ArcSight environment to scale, and opens up ArcSight events to third-party data solutions.
ArcMC	HPE ArcSight Management Center (ArcMC) is a centralized management tool that simplifies security policy configuration, deployment maintenance, and monitoring efficiently and cost-effectively. ArcMC provides run-time management of Event Broker topics. ArcMC is sold as part of the ArcSight Deployment Platform (ADP).

Event Broker manages the distribution of events in topics to which consumers can subscribe.

- The CEF version configured at the Connector on the producer side (CEF 0.1 or 1.0) should be the CEF version supported by the consumer.
- There are two EB topics you can configure your SmartConnector to connect to: eb-cef and eb-esm. The ESM topic produces binary, which is the format ESM consumes.
- Multiple connectors can be configured to publish to the eb-cef topic. Load is balanced by EB.
- Event Broker's stream processor converts CEF-formatted event data in the eb-cef topic to Avro format and sends the Avro formatted event data to eb-internal-avro topic.
- ArcSight ESM can be configured as an Event Broker consumer (consuming data from the eb-esm topic).

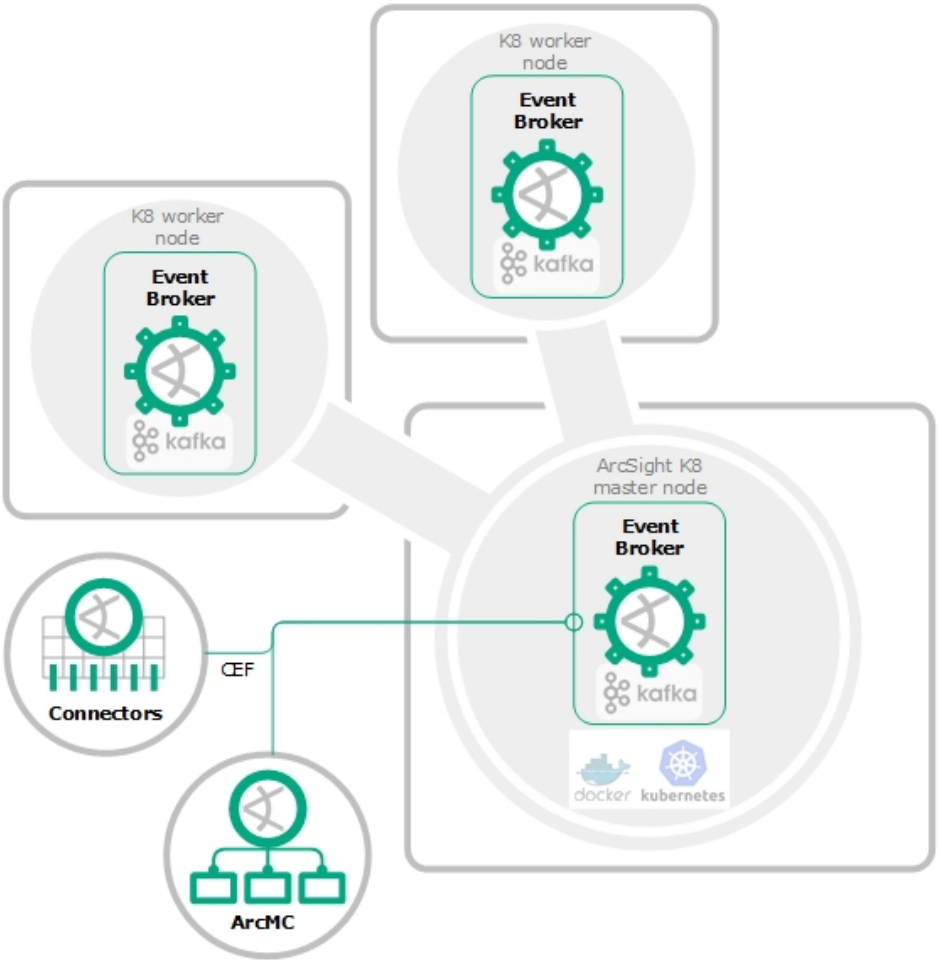
Supported deployment scenarios

ArcSight Event Broker supports the following deployment scenarios:

	Use Case	Description	Guidance document
1	Existing ADP 2.0 customer with EB 1.0 adding ArcSight Investigate	1. Upgrade EB 1.0 to 2.0, and migrate EB 1.0 data to EB 2.0. Check with Customer Support for the availability of data migration instructions. 2. Upgrade ArcMC to 2.6 or 2.6.1 3. Install Investigate from ArcSight Installer app	1. EB data migration tech note 2. ArcMC Administration Guide 3. Investigate Deployment Guide
2	Stand-alone EB 2.0 w/o Investigate	Fresh install using stand-alone EB version of ArcSight Installer app. EB to be managed by ArcMC 2.6 or later.	Event Broker Administration Guide ArcMC Administration Guide
3	Stand-alone EB 1.0 to EB 2.0 data migration	Instructions for migrating data from EB 1.0 to EB 2.0	EB data migration tech note
4	Upgrade EB 2.01 to EB 2.02	Upgrade an existing installation of Event Broker to Event Broker 2.02	Event Broker Deployment Guide

ArcSight Event Broker deployment architecture

ArcSight Event Broker is installed using Docker containers managed by Kubernetes and deployed from the ArcSight Installer application. The default deployment consists of a Kubernetes master node, and two Kubernetes (k8s) worker nodes.



Deployment component	Host	Functional contents
Kubernetes master node	1 VM or physical server	• Kubernetes master node • Kafka Event Broker node
Kubernetes worker nodes	2 VMs or physical servers	• 2 Kafka Event Broker nodes
ArcSight SmartConnectors	Stand-alone or hosted on a Connector-Hosting Appliance	Normalizes event data from network devices and formats as CEF.

Deployment component	Host	Functional contents
ArcSight Management Console	Separate installation	Provides run-time management of Event Broker topics.

About how Kubernetes nodes are managed

ArcSight Event Broker is installed and deployed by the ArcSight Installer application using Kubernetes container management to enable elastic scaling. The Kubernetes master node controller resides on one system/node. A Kubernetes worker node hosts container management units called pods. A pod manages one or more containers with a shared namespace and shared volumes.

Default system pods	Event Broker pods + containers	ArcSight Investigate pods + containers
default-http-backend-*	eb-c2av-processor-* (pod) c2av-processor (container)	hercules-management-* (pod) kubernetes-vault-renew (container) hercules-management (container)
nginx-ingress-controller-*	eb-kafka-0 (pod) kafka (container)	hercules-rethinkdb-0 rethinkdb (container)
	eb-kafka-1 (pod) kafka (container)	hercules-search-* (pod) kubernetes-vault-renew (container) hercules-serach-engine (container) hercules-search (container)
	eb-kafka-2 (pod) kafka (container)	

Default system pods	Event Broker pods + containers	ArcSight Investigate pods + containers
	eb-kafka-manager-* (pod) kafka-manager (container)	
	eb-routing-processor-* routing-processor (container)	
	eb-schemaregistry-* (pod) schemaregistry (container)	
	eb-web-service-* (pod) kubernetes-vault-renew (container) atlas-web-service (container)	
	eb-zookeeper-0 (pod) zookeeper (container)	
	eb-zookeeper-1 (pod) zookeeper (container)	
	eb-zookeeper-2 (pod) zookeeper (container)	

During setup and deployment, the pods with an asterisk (*) are appended with a randomly-generated identification number.

The Kafka worker nodes provide redundancy and load balancing. If one node fails or is taken offline, the others take over.

You can add nodes to the default configuration for Kafka or ZooKeeper. Adding nodes requires a change to the installer.properties file and additional resource planning. For details, see Adding Kubernetes nodes.

ArcSight Event Broker deployment overview

1. ArcSight Event Broker prerequisites
2. Install ArcSight Event Broker using the ArcSight Installer application
3. Configure ArcSight Event Broker components
4. Adding Kubernetes nodes
5. Uninstalling ArcSight Event Broker
6. ArcSight Event Broker deployment troubleshooting and FAQs

What's next

ArcSight Event Broker prerequisites

ArcSight Event Broker prerequisites

System requirements

The information in this topic provides general sizing guidelines based on a default setup. For tailored sizing recommendations for a production environment, contact ArcSight Customer Support.

Provision the servers (or VMs) you're using for the deployment with the following. For supported platforms and operating systems, see the ADP Support Matrix in the documentation space of Protect 724.

Component	# Nodes	Resources needed	Available ports
Event Broker	1 master 2 worker nodes	• One CPU with 24 cores • 32 GB RAM • 8 TB disk space • RHEL or CentOS 7.3 • Linux kernel version 3.10 or higher • 10 GigE network or equivalent	32181, 9093, 9092
ArcMC (part of ADP)	1	• One CPU quad-core • 16 GB RAM • 50 GB of free disk space For ArcMC deployment details, see the *ArcMC Administrator's Guide*.	
SmartConnectors (part of ADP)	1	SmartConnector version 7.6 (can be stand-alone or managed by ArcMC) For ArcSight SmartConnector deployment details, see the *SmartConnector User's Guide*.	

Preparing producer/consumer interfaces and encryption modes

Follow these guidelines for preparing optional producer/consumer components. See the indicated guidance documentation for detailed instructions. For instructions about how to configure producers and consumers after ArcSight Event Broker deployment, see Configure ArcSight Event Broker components.

Set up encryption modes before installing and configuring Event Broker

Before installing Event Broker, determine the encryption mode you want to use to encrypt communications between ArcSight components. Set up the other ArcSight components with the encryption mode you intend to use before connecting them to the Event Broker. The security mode of systems connected to EB (Consumers, Producers, ArcMC) must be the same as the security mode set for Event Broker. Changing encryption modes after Event Broker has been deployed will require system down time. If you do need to change the security mode after Event Broker deployment, see the Event Broker Administrator's Guide.

Product	**Preparations needed**	**Open ports**	**Supported encryption modes**	**Guidance documentation**
ArcMC	Install ArcMC before ArcSight Event Broker installation.	38080	• TLS • FIPS • ClientAuth	ArcMC Administrator's Guide

Product	Preparations needed	Open ports	Supported encryption modes	Guidance documentation
ArcSight SmartConnectors	ArcSight SmartConnectors and ArcMC onboard connectors can be installed and running prior to installing ArcSight Event Broker. FIPS mode setup is not supported between Connector version 7.6 and Event Broker. TLS and ClientAuth are the only encryption methods supported between SmartConnector version 7.6 and Event Broker.	9093	• TLS • FIPS • ClientAuth	SmartConnector User Guide ArcMC Administrator's Guide
ArcSight ESM (optional)	ArcSight ESM can be installed and running prior to installing ArcSight Event Broker.	9093	• TLS • FIPS • ClientAuth	ESM Installation Guide ESM Administrator's Guide
ArcSight Logger (optional)	ArcSight Logger can be installed and running prior to installing Event Broker.	9093	• TLS • FIPS • ClientAuth	Logger Administrator's Guide

Preparations for systems managed by Kubernetes

For all systems that will be managed by Kubernetes, do the following:

1. Prepare the Kubernetes master and worker nodes using the guidelines provided above.
2. The nodes require public key encryption to communicate.
    a. To generate a key pair, run the following command on the master node:
    ssh-keygen -t rsa
    b. Run the following command on the master node to copy the public key to the worker nodes:
    ssh-copy-id -i ~/.ssh/id_rsa.pub root@<workernode_hostname>
3. Configure NTP using Chrony on all of the hosts in the cluster.
    Chrony is installed by default on some versions of Red Hat and CentOS. However, if you do not have Chrony installed, run the following commands:
    a. Install Chrony:

    ```
    yum install chrony
    ```

    b. Start Chronyd:

    ```
    systemctl start chronyd
    systemctl enable chronyd
    ```

    c. Verify that Chrony is operating correctly:

    ```
    chronyc tracking
    ```

TLS planning

The various components in the ArcSight Event Broker system interact using encrypted communication implemented using TLS 1.2 protocol.

TLS implementation requires digital certificates. Before you begin the installation process, you must decide on the type of certificate you prefer to use:

- Kubernetes self-signed certificate. Kubernetes includes the capability to generate self-signed certificates. By default, the Kubernetes installation process generates certificates for the Kubernetes cluster, but you can instruct otherwise during the installation process. You can also generate a Kubernetes certificate for other components in the system, which require a certificate, like the ArcSight Investigate Vertica database. For more information on generating a Kubernetes certificate, see Generate a signed certificate from the system CA.

- A valid digital certificate signed by a certificate authority (CA). Depending on your organization's security policy, you might be required to use a certificate from a trusted CA. In this case, make sure that you have a root certificate file and a private key file. Copy these files to the designated Kubernetes master node.

Note: The certificates cannot be reconfigured after installation.

If you are planning on enabling FIPS mode, make sure the certificate generated meets the FIPS criteria.

Preparations for all systems

On all servers, master and worker nodes, do the following. All commands must be run as a root user.

- Firewall should be enabled. (The ArcSight Installer opens the needed ports.) Run the commands `systemctl enable firewalld and systemctl start firewalld.`
- Enable internet access in order to download the product container images.
- If your organization's network has a proxy, define the proxy environment variable on all servers.
- Reboot all master and worker nodes.
- Ensure that each node is configured with a fully qualified domain name
- Ensure proper DNS configuration across all systems including correct forward and reverse proxy lookups.
- Disable SELinux.
- Increase the default user process (ulimit), as follows:

    a. Open the file /etc/security/limits.d/*-nproc.conf.
        1. If you do not already have a /etc/security/limits.d/*- nproc.conf file, create one (and the limits.d directory, if necessary).
        2. If the file already exists, delete all entries in the file.
    b. Add the following lines. Be sure to include the asterisk * in the new entries. It is important that you add all of the entries exactly as specified.

```
* soft nproc 10240
* hard nproc 10240
* soft nofile 65536
* hard nofile 65536
```

Set up proxy server

If you use a proxy server in your environment, define a proxy environment variable on all servers.

Procedure

1. Configure proxy settings (if needed) to get yum access to CentOS repository.

```
cat >> ~/.bashrc <<EOL
export ftp_proxy=<your_proxy_url:port>
export http_proxy=<your_proxy_url:port>
export https_proxy=<your_proxy_url:port>
EOL
source ~/.bashrc
```

2. Reboot all Kubernetes master and worker nodes. Nodes can be rebooted in any order.

Installation Order

You can install ArcMC and ArcSight SmartConnectors in any order during ArcSight Event Broker deployment, however, ArcSight recommends having them installed and running in your environment before installing the Event Broker components. Additional configuration of both components will be required after you install ArcSight Event Broker.

What's next

Install ArcSight Event Broker using the ArcSight Installer application

Install ArcSight Event Broker using the ArcSight Installer application

Before running the ArcSight Installer application, verify that you have set up the receiving systems according to guidelines in Event Broker prerequisites. The ArcSight Installer will configure firewall settings during setup (in case firewalld.service is up and running) on both the Kubernetes master and worker nodes. Multi-master installation is not supported.

This procedure provides instructions for installing online using the Docker Hub repository, or offline by downloading a tar file from an FTP site and replicating a local Docker Hub on the master node system.

Generate keypair on master node for worker nodes

In a master/worker node deployment, generate a key pair on the master node and copy the public key to each worker node. This enables passwordless SSH access from the master server to all the other worker node servers in the cluster. Do this before you install the ArcSight Installer, and before you install and setup Kubernetes.

The following is an example of enabling passwordless SSH (You can find other examples online like this one http://www.linuxproblem.org/art_9.html)

Note: Generate the keypair as the root user.

Procedure

1. Run the ssh-keygen command on the master server, for example:

```
ssh-keygen -q -t rsa
```

2. Enter a file in which to save the key. For example:

```
/root/.ssh/id_rsa
```

3. Enter a passphrase. Enter the passphrase again.

4. Copy the key from the master node to the worker node using the worker node's IP address. For example:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@<worker node IP>
```

5. The system displays the key fingerprint and requests to authenticate with the worker node server. Enter the worker node passphrase credential as required. The operation is successful when the system displays the following message:

```
Number of key(s) added: 1
```

6. To verify that the key was successfully installed on the worker node, run the following command from master to the worker node to verify that it can successfully log into the worker node.

```
ssh root@<worker node IP>
```

7. Repeat steps 4 through 6 for every worker node.

Install ArcSight Installer on the master node

The following procedure installs the ArcSight Installer, starts the application, and unpacks the Kubernetes scripts to /opt/arcsight/installer/k8s.

Note: Perform the installation as a root user or sudo user.

Procedure

1. Copy the installation package and the md5 checksum file to the master server.

2. Verify the integrity of the installation package. Run md5sum <installer filename> and cat <installer filename>. These two sums should be identical.

3. On the master server, change directories to the location of the installation package, and then run the following command; for example, "arcsight-installer-x.x.x.x86_64.rpm".

```
yum install -y <rpm_name_version>.rpm
```

Install and set up Kubernetes

The following procedure deploys Kubernetes on the master node and sets up the worker nodes.

Note: Perform the installation as a root user.

Procedure

1. Run the following script on the master server.

```
sh /opt/arcsight/installer/k8s/master/install.sh [optional parameters]
```

Optional Parameters

a. To update the default configuration of the Docker log rotation, use the following parameters. (By default, Docker log rotation keeps a total of 5 files per container, each file with a maximum size of 20 Mb.)

LOG_MAX_SIZE: specifies the maximum log file size, per container, for Docker log rotation, indicating a number followed by the unit of measure (k=Kilobytes, m=Megabytes, and g=Gigabytes).

LOG_MAX_FILE: specifies the maximum number of files to keep, per container, for Docker log rotation. Example for LOG_MAX_SIZE and LOG_MAX_FILE:

```
sh /opt/arcsight/installer/k8s/master/install.sh LOG_MAX_SIZE=100m
LOG_MAX_FILE=5
```

b. If you are using a certificate from a trusted CA, use the following parameters.

ROOTCA: The root or intermediate certificate for generating client and server certificates.

ROOTCAKEY: The private key for generating the client and server certificates.

CLOUD_PROVIDER: This parameter is required if your cloud provider is Microsoft Azure, which requires a specific configuration. Pass the following value: azure.

Example for ROOTCA and ROOTCAKEY:

```
sh /opt/arcsight/installer/k8s/master/install.sh ROOTCA=/tmp/ca.crt
ROOTCAKEY=/tmp/ca.key
```

Example for CLOUD_PROVIDER

```
sh /opt/arcsight/installer/k8s/master/install.sh CLOUD_PROVIDER=azure
```

Note: If a proxy is to be used, and you are not performing offline installation, enter the proxy in the very next step, at the blank prompt, in the following format: `http://proxy.example.com:80/. (`The blank prompt is easy to miss.)

If you are performing offline installation, at the blank prompt, press enter to skip the blank prompt. There is no need to enter any information at this step.

2. Run the following command on the master server for each of the worker nodes. Worker node IP should be specified in an IPv4 format (1.1.1.1).

```
sh /opt/arcsight/installer/k8s/node/install.sh -w <worker node IP>
```

3. Verify that the ArcSight Installer was installed successfully. Open the ArcSight Installer URL in a browser on the master server. The ArcSight Installer login screen will load in the browser. If you are using self-signed certificates, accept the self-signed certificate notification from the browser.

```
https://<kubernetes_master>:8888/
```

4. Create a new cluster.

    a. Open the ArcSight Installer on the master server:

```
https://<kubernetes_master>:8888
```

    b. On the Login page click **Create New Cluster.**
    c. On the **Add New Cluster** page, enter the following information, and click **Create**:

        - **Cluster ID**. Enter an ID made up of letters or letters and numbers or underscore character, and must be at least 5 characters in length. Spaces are not supported.
        - **Password**/**Confirm Password**. Enter a password that is at least 6 characters long, and confirm it.

5. Label all nodes with the kubectl label command. (The commands shown below will label a 3-node setup of 1 master and 2 workers. Apply labels to every node in your cluster `<workerN_ip>` based on your own setup, as needed.)

```
kubectl label --overwrite node <master>

kubectl label --overwrite node <worker1_ip> zk=yes
```

```
kubectl label --overwrite node <worker1_ip> kafka=yes

kubectl label --overwrite node <worker2_ip> zk=yes

kubectl label --overwrite node <worker2_ip> kafka=yes

…

kubectl label --overwrite node <workerN_ip>  zk=yes

kubectl label --overwrite node <workerN_ip> kafka=yes
```

6. Set up the connections to the master Kubernetes node and Docker Hub.

    a. In the ArcSight Installer, on the **Cluster Setup** page, click **Edit** next to the **Master** field.
    b. On the **Master Configuration** page, enter the following information, and click **Save**:

    - **MasterAddress**. Enter the hostname or IP address of the master server.
    - **Token**. Copy the content of the token before the first comma (,) character (such as XXXX from XXXX,admin,admin) into this field from the following location in the master node:
      /opt/arcsight/kubernetes/ssl/token
    - **CA Certificate**. Copy the content of the CA certificate into this field from the following location in the master node:
      /opt/arcsight/kubernetes/ssl/ca.crt

7. Set the Docker Hub configuration. On the Cluster setup page > Docker Repository field, click **Edit** to set the Docker Hub configuration.

    a. If you are installing offline, see the section on offline installation instructions below–in URL, enter 127.0.0.1:5000, and use dummy data for UserName, Password, and Email. Email must be a valid email format.
    b. If you are installing online, fill in the dialog with the URL for the Docker Hub: index.docker.io, and use your account-specific UserName, Password, and Email.

8. Verify that the Kubernetes master node was installed successfully:

    a. In the ArcSight Installer, click **Node Management** and verify that the servers that you installed are **READY**.

Note: Allow several minutes for the established nodes to appear as **READY** on the Node Management page.

If the nodes are not ready after several minutes, refresh the page.

.

Status	Description
NOT_READY	The node is available but not ready for product deployment.
READY	The node is available and ready for product deployment.
ERROR	The node is available, but cannot be deployed due to a system error. Hover over the error icon to see the message.

Adjust installer.properties as needed before deployment

Before deploying the nodes, adjust properties as needed for your environment. You would need to adjust the properties set here if you are deploying in FIPS mode, or want to add more worker nodes to the default configuration.

Property file setting	Default value	Options
All Event Broker components will use FIPS-certified encryption algorithms	predeploy.eb.init.fips=false	Set to true if you are setting up FIPS encryption
Event Broker Kafka will use TLS Client Authentication to verify client connections	predeploy.eb.init.client-auth=false	Set to true if you are setting up client-auth authentication
Number of partitions for Event Broker	predeploy.eb.init.noOfTopicPartitions=6	Set a different number of

Property file setting	Default value	Options
topics in Kafka		partitions as needed
Replication factor for Event Broker topics in Kafka	predeploy.eb.init.topicReplicationFactor=2	Set a different replication factor for Kafka as needed
Kafka log retention size (bytes)	predeploy.eb.init.kafkaRetentionBytes=10737418240	Set a different size (in bytes) for the Kafka log retention size as needed
Kafka log retention size for the Vertica avro topic. This is uncompressed and requires more space to hold events for the same duration. (bytes)	predeploy.eb.init.kafkaRetentionBytesForVertica=10737418240	Set a different size (in bytes) for the Vertica Avro topic (if present) as needed
Kafka log retention duration (hours)	predeploy.eb.init.kafkaRetentionHours=672	Set a different duration for the Kafka log retention as needed
Kafka inter-broker	predeploy.inter.broker.protocol.version=0.10.1.0	Only modified

Property file setting	Default value	Options
protocol version		during upgrades.
The message format version the broker will use to append messages to the logs.	predeploy.log.message.format.version=0.10.1.0	Only modified during upgrades.
Number of Kafka and ZooKeeper nodes	predeploy.eb.kafka.count=3 predeploy.eb.zookeeper.count=3	
Host path to store data persistently	predeploy.eb.kafka.path=/opt/arcsight/k8s-hostpath-volume/eb/kafka predeploy.eb.zookeeper.path=/opt/arcsight/k8s-hostpath-volume/eb/zookeeper	
ArcMC hostname	predeploy.eb.arcmc.hosts=localhost:443	Specifies location of the managing ArcMC.
The endpoint identification algorithm to validate the server hostname using the server certificate.	predeploy.ssl.endpoint.identification.algorithm=	
The number of stream threads	predeploy.stream.num.threads=6	

Property file setting	Default value	Options
Truncate fields in C2av	predeploy.c2av.field.truncate=false	For EB + ArcSight Investigate. Change to true to avoid events being sent to Vertica's reject table in case the event field's information size exceeded the defined length.
Log level for each EB container	predeploy.level=info predeploy.kafka.log.level=${predeploy.level} predeploy.zookeeper.log.level=${predeploy.level} predeploy.schema.log.level=${predeploy.level} predeploy.web.service.log.level=${predeploy.level} predeploy.c2av.stream.processor.log.level=${predeploy.level} predeploy.eventbroker.routing.processor.log.level=${predeploy.level}	For debugging, change "info" to "debug"
Host path directory for ArcMC certificates	predeploy.arcmc.certs.path=/opt/arcsight/k8s-hostpath-volume/eb/arcmccerts	This setting is populated during Event Broker setup in ArcMC.

Offline download and local Docker Hub setup instructions (before deployment)

This procedure applies if you do not have an Internet connection from your ArcSight Event Broker servers to the HPE Docker Hub registry.

Procedure

1. From a server with an Internet connection, connect to the HPE Software Entitlements portal and download the offline installer files:

Component	Offline installer file name
ArcSight Installer application	arcsight-installer-1.10.7-ga110.x86_64.rpm
ArcSight Event Broker	arcsight-eventbroker-2.02.0-images_66_4f7e26.tar
ADP Event Broker (ADP EB only)	arcsight-installer-1.10.7-ga110_eb.x86_64.rpm

2. Verify the integrity of the files using the Docker inspect command. Event Broker includes 6 Event Broker images. In the command below, .Id represents 'Id' field of the image, which is the unique identifier of a Docker image. Run the following commands and compare your output with what is shown. Your output should be the same.

```
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_kafka-manager:2.02.0
sha256:b1849f64a427e64c896b9070b708a48f053bda4828b04638bab2adce41cc0f48
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_web-service:2.02.0
sha256:b0e0b734ed54ca02f49f927a1e0af6e43423f8f5a91effff03ecaac11ed8a57b
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_sp:2.02.0
sha256:5d51459969571501a798d9d7ba798b579da44fd9aa98d5892a4ac3505f444ca9
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_schema-registry:2.02.0
sha256:d79b406bda0573489d185bb587db567d564a18047204c5474638dbb0066c6583
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_kafka:2.02.0
sha256:4e8bcf9899e02c741cae1aa21825d77da3bf3c29061b49bc250bb29d7bb9ea9d
# docker inspect --format='{{.Id}}' arcsightsecurity/atlas_zookeeper:2.02.0
sha256:cb96c4e0e960e26b2ba0b66e8ed195df9d1a1024fbdb312e52d23c9b751da527
```

3. Copy the rpm and tar files to any location on the master node.

4. Push the images to the private local registry. A private Docker registry is configured and running on 127.0.0.1:5000 and is accessible from all nodes. Run the following command on the master node:

```
/opt/arcsight/installer/k8s/master/pushImages.sh -f <images.tar>
```

5. In the ArcSight Installer, on the **Cluster Setup** page, click the **Edit** button next to **Docker Repository**.

6. On the **Docker Hub Configuration** dialog box, in the **URL** field enter the following URL:

```
127.0.0.1:5000
```

7. Click **Save**.

Deploy Event Broker worker nodes in the ArcSight Installer

Procedure

1. In the ArcSight Installer, click the **Deployment** tab.

    a. If you are doing an online installation using a direct connection to Docker Hub, proceed to step 2.

    b. If you are doing an offline installation using the downloaded rpm/tar files and the local Docker Hub on 127.0.0.1:5000, be sure you pushed the images to the local Docker Hub repository as outlined above in Offline download and local Docker Hub setup instructions.

2. Click **Deploy** next to Event Broker.
   You can monitor the deployment status in the status column. The initial status is **IN_PROGRESS**. Deployment may take a while depending on the connection speed to the Docker Hub repository. When the products are deployed, the status changes to **DEPLOYED**.

Status	Description
NOT_READY	ArcSight Installer cannot communicate with Kubernetes. Deployment is not possible.
OFF	ArcSight Installer is able to communicate with Kubernetes. The products are not deployed yet.

Status	Description
IN_PROGRESS	deployment or undeployment has started and is in progress. This status can also display when one or more containers get restarted on Kubernetes (for example, when you change product configurations).
DEPLOYED	the product is successfully deployed and running.
ERROR	one or more product containers is broken. Hover over the error icon to see the message. This status may show up and then turn into DEPLOYED when a container has crashed and then fixed or restarted by Kubernetes.

Configure ArcSight Event Broker components

Configure Event Broker for management by ArcMC

- Make sure EB host can be identified using IP address
- Register EB host with ArcMC
- Manage nodes in ArcMC

See the ArcMC 2.6 Administrator's Guide topic "Adding a host" for details.

Configure SmartConnectors

- Set Event Broker destinations appropriate for your topology in the SmartConnector Configuration screen. For details, see the SmartConnector User's Guide.
- Make sure events are coming in and returned in search.
- Add more connectors as desired.

Adding Kubernetes nodes

Adding worker nodes is supported. Once the new worker nodes are added, labels can be used to assign specific pods to them, like with ZooKeeper and Kafka. This procedure describes how to add worker nodes to extend the Kafka cluster nodes.

1. To add a new Kafka node add a new worker node and label it, update the replica count and update the installer properties. For example, if you want to add two more nodes to an existing 3-node Kafka cluster to create a 5-node Kafka cluster:

   /opt/arcsight/installer/k8s/node/install.sh –w <new_worker_node_1_IPv4_address>

   /opt/arcsight/installer/k8s/node/install.sh –w <new_worker_node_2_IPv4_address>

   kubectl label --overwrite node <new_worker_node_1_IPv4_address> zk=yes kafka=yes

   kubectl label --overwrite node <new_worker_node_2_IPv4_address> zk=yes kafka=yes

   kubectl scale petset eb-kafka --replicas=5

2. Update installer.properties with the new kafka node count so future deploy/undeploy will have the correct kafka node number.

   $ vi /opt/arcsight/installer/installer.properties

   …

   predeploy.eb.kafka.count=5

   …

   $

Generate signed certificates for consumers

Event Broker consumers need a signed certificate from the Event Broker to establish secure communication.

There are a number of methods for generating signed certificates. Each used for different use cases:

- ArcSight Installer includes a utility for generating a signed certificate from the CA that is configured in the system (either Kubernetes or another trusted CA). You can use this utility to generate certificates for other components in the system, such as the ArcSight Investigate Vertica database.
- ArcSight Installer includes a utility for generating a signed certificate from a Certificate Signing Request (CSR) file. You can use this utility to generate certificates for client authentication, such as ESM and logger.

Note: You can perform these procedures only after Kubernetes is installed.

Generate a signed certificate from the system CA

Procedure:

1. Run the following command on the Kubernetes master server:

sh /opt/arcsight/installer/k8s/master/cert-utils.sh generate-certificate

The following argument is required:

- $1: FQDN - fully qualified domain server name

The following files are created in the directory where you ran the command:

- <FQDN>.crt
- <FQDN>.key

2. Copy the files to the server for which you generated the certificate.

Generate a signed certificate from a CSR

Procedure:

1. Copy the CSR file to the Kubernetes master server.

2. Run the following command on the Kubernetes master server:

sh /opt/arcsight/installer/k8s/master/cert-utils.sh sign-certificate-request

The following arguments are required:

- $1: The CSR file (full path).
- $2: The name of the CRT you want to create (without the crt extension).

A certificate and a private key are created

3. Copy the files to the server for which you generated the certificate.

Uninstalling ArcSight Event Broker

To uninstall you must perform the following procedures in this order:

1. Uninstall ArcSight Installer.

2. Uninstall Kubernetes.

Procedure

1. Uninstall ArcSight Installer. Run the following commands on the master server in this order:

   yum erase -y arcsight-installer.x86_64 to uninstall Arcsight Installer

   rm -rf /opt/arcsight/installer to remove all log files and directories left under /opt folder

2. Uninstall Kubernetes.

   Run the following command on all the worker nodes and on the master server and reboot:
   sh /opt/arcsight/kubernetes/uninstall.sh

Optionally, rm -rf /opt/arcsight/ on master and all worker nodes to delete all the data created by Event Broker. (This will delete all events and configuration files.)

Warning: Do not reinstall Event Broker on a system without deleting the data files. If not deleted, then Event Broker will come up using old data files and not function properly.

Upgrading Event Broker

Follow these steps to upgrade Event Broker to version 2.1.

1. Stop the ArcSight Installer service.

```
# systemctl stop arcsight-installer
```

2. Navigate to the Kubernetes directory and store the ca.key and ca.crt in a secure location.

```
# cd /opt/arcsight/kubernetes/ssl/

# cp -r ca.* /tmp
```

3. Uninstall Kubernetes on the master and any worker nodes:

```
# ../sh uninstall.sh

# ssh root@<worker_node IP> 'sh
/opt/arcsight/kubernetes/uninstall.sh'
```

4. Remove the Arcsight Installer with yum:

```
# yum erase arcsight-installer
```

5. Download the latest version of the ArcSight Installer and associated md5 checksum file. Verify the md5checksum, and then install the file.

```
# yum install arcsight-installer-<build>.x86_64.rpm
```

6. Install the latest Kubernetes files using the keys you retained from Step 2.

```
# cd /opt/arcsight/installer/k8s/master/

# sh install.sh ROOTCA=/tmp/ca.crt ROOTCAKEY=/tmp/ca.key

# kubectl label node <master node IP> zk=yes kafka=yes

# cd /opt/arcsight/installer/k8s/node

# sh install.sh -w <worker node 1 IP>

# kubectl label node <worker node 1 IP> zk=yes kafka=yes

# sh install.sh -w <worker node 2 IP>

# kubectl label node <worker node 2 IP> zk=yes kafka=yes
```

7. Navigate to the Arcsight Installer web home page. In the Installer, update the token only (certificate is the same), and enter the Docker credentials. To get the new token content, `cat`

`/opt/arcsight/kubernetes/ssl/token` and copy everything before the first comma (,) character (such as XXXX from XXXX,admin,admin).

8. Deploy Event Broker and other updated products as needed.
9. Configure Event Broker settings for the new version.

Offline Upgrade

For an offline upgrade of Event Broker, follow these steps.

1. Stop the ArcSight Installer service.

```
# systemctl stop arcsight-installer
```

2. Navigate to the Kubernetes directory and store the ca.key and ca.crt in a secure location.

```
# cd /opt/arcsight/kubernetes/ssl/

# cp -r ca.* /tmp
```

3. Uninstall Kubernetes on master and any worker nodes:

```
# ../sh uninstall.sh

# ssh root@<workNodeIP> 'sh /opt/arcsight/kubernetes/uninstall.sh'
```

4. Remove the Arcsight Installer with yum:

```
# yum erase arcsight-installer
```

5. Download and install the latest version of the ArcSight Installer and the latest offline .tar files.

```
# yum install arcsight-installer-<build>.x86_64.rpm
```

6. Install the latest Kubernetes files using the keys you retained from Step 3. If prompted for a proxy server, just press Enter.

```
# cd /opt/arcsight/installer/k8s/master/

# sh install.sh ROOTCA=/tmp/ca.crt ROOTCAKEY=/tmp/ca.key

# kubectl label node <master node IP> zk=yes kafka=yes

# cd /opt/arcsight/installer/k8s/node

# sh install.sh -w <worker node 1 IP>

# kubectl label node <worker-node 1 IP> zk=yes kafka=yes

# sh install.sh -w <worker node 2 IP>
```

```
# kubectl label node <worker node 2 IP > zk=yes kafka=yes
```

7. Navigate to the Arcsight Installer web home page (https://<home IP>:8888). Use the same credentials as before to log in.
8. In the Installer, update the token only (certificate is the same). To get the new token content, `cat /opt/arcsight/kubernetes/ssl/token` and copy everything before the "," character (such as XXXX from XXXX,admin, admin).
9. Log in using the following:

```
URL 127.0.0.1:5000

Username: <as previous>

Password: <as previous>

Email: <as previous>
```

10. Push the new images for Event Broker, and Investigate (if needed).

```
cd /opt/arcsight/installer/k8s/master

./pushImages.sh -f /opt/arcsight_investigate_<file id>.tar

./pushImages.sh -f /opt/ arcsight_eb__<file id>.tar
```

11. Configure Event Broker settings for the new version.
12. Deploy Event Broker and Investigate from the ArcSight Installer UI.

Next steps

After upgrade, you may need to take these additional steps on other systems:

ArcMC: If your Event Broker was previously managed by ArcMC before the upgrade, the ArcMC Monitoring Summary will not reflect the correct Event Broker information for the upgraded version until you import a new Event Broker certificate into ArcMC. In addition, any managed hosts that are consumers of Event Broker (such as managed Loggers) must also receive a new certificate.

In the *ArcMC Administrator's Guide,* see Downloading and Importing Host Certificates for more information.

Logger: If you have Logger configured as an Event Broker consumer, after the upgrade, disable and then enable the Logger Receiver.

ArcSight Event Broker deployment troubleshooting and FAQs

Troubleshooting

Where to find the logs

Use the following command to access the logs: `kubectl logs eb<podname>`

Pod starting order

After deploying EB, pods are configured to start in the following order. Downstream pods will not deploy until the dependencies are met.

1. A quorum of ZooKeeper pods in the cluster must be up (2 of 3, or 3 of 5). 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, Kafka Manager
5. Transformation Stream Processor, Routing Stream Processor

Pods Restarting During First Deployment

During the very first deployment of Event Broker It is normal to see some of the pods restarting (like the schema registry pod) while all the required Docker images for all event broker pods are being downloaded from the Docker hub for the first time. When deploying Event Broker, pods are configured to start in a specific order and the downstream pods will not deploy until the dependencies are met and will restart after certain wait time and number of retires.

The Docker image download depends on your network download speed and configuration.

Cannot query ZooKeeper

Symptom: when you run kubectl get pods command to get status of the pods and you see that downstream pods (see the pod star order) do not stay up and the status is a 'CrashLoop'-type error.

Conditions to look for:

- Check that ZooKeeper pods are running.

- If the ZooKeeper pod status is Pending, you may not have labeled the nodes (zk=yes). Verify that the nodes are labeled using the kubectl get nodes -L=zk command.

- Verify that you configured an odd number of ZooKeeper in installer.properties predeploy.eb.zookeeper.count attribute.

- Check the zookeeper pod logs for errors using the kubectl logs <pod name>.

Common Errors/Warnings in ZooKeeper Logs

- Quorum Exceptions: Cannot elect a leader. If you see this type of error, check the check conditions above.

- Socket errors: this can occur if there are too many connections. The solution is to restart the pod using the kubectl delete <pod_name>. The pod will be recreated automatically.

Common Errors/Warnings in Kafka logs

- You see indicating that broker Cannot Register ID: It might cause by multiple brokers with the same ID. This is a rare situation that can occur when you are add and removing nodes from the cluster and you do not define the cluster properly.
  How to verify whether this is an issue: Connect to each system that is running a Kafka broker and check the assigned broker.id value of each. The broker.id value defined on each Kafka node must be unique.

  # cat /opt/arcsight/k8s-hostpath-volume/eb/kafka/meta.properties

  version=0
  broker.id=1001

SSL Connection Error

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

- Cannot communicate with other brokers: If you see this type of error, host names may not be configured properly. It is possible that the node cannot perform reverse look up or that DNS is not set up properly.

- Out of disk condition (logs or data): <need to provide the actual error>: It is important to configure retention size correctly so that this does not occur. If the error occurs on one node, and other nodes are operating with no problems, you can delete the files (the other nodes in the cluster will continue to handle the event flow), change retention size on the topic manually, then restart the pod.

  One option is to add more nodes to the cluster. Each node will store less data. Contact support or services if you want to take this approach.

  Example: A Kafka node goes down, then the Schema Registry goes down. The Schema Registry will not restart while one Kafka node is down since it requires that all kafka nodes are up.

- First, work to get the Kafka node back up. the Schema Registry pod requires that all Kafka pods are running. The Schema Registry will continue to check whether it's dependency pods are running until it

they are met. If Schema Registry crashes while performing this check, then Kubernetes will restart the pod and Schema Registry will continue to check.

- Consequence: The Transformation Stream Processor (C2AV) will not function while SR is down. If SR is down for an extended period of time, the message queue in the eb-cef topic increase as because messages are not being processed (converted to AVRO). As long as the topic retention policy is set large enough, this should cause a problem. The message queue for the cef topic will continue to increase without events being deleted. If the topic size or time range reaches the retention policy, then older messages will be deleted from the topic.

- If the Kafka node is permanently down, you might be able to edit the yaml file for Schema Registry (this is an advanced task; must be done by services or support.)

The EB pre-defined topics are not created on the initial EB first deployment

- Symptom: the Bootstrap Web Service log contains 500 response code (the response from SR), and topics are not created.

- Work around: Undeploy Event Broker containers, and then redeploy them.

One or more connectors cannot send data to Kafka

- Check whether the connection configuration is set properly in the connector.

- Check that the encryption mode (TLS, TLS+FIPS, TLS+CA, TLS+FIPS+CA) is the same for the Connector and Event Broker.

- Make sure you can connect to the Kafka port on the system and that there are no network issues.

- If you encounter a certificate error (“cannot retrieve the certificate”) 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 able to sent through EB. You need to fix the broker issue to that it comes up. In general, topics should be configured with replication > 1 so to prevent this scenario.

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

- Check whether the disk full.

Vertica cannot read events from Kafka

- New set up: Vertica Kafka scheduler: Check that kafka scheduler is configured to communicate to Kafka port 39092.

- Working at first, but stopped working: Offset is not 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 meta data. After doing this, immediately run the kafka_scheduler create command to set up the scheduler.

- New set up: Check the network connection.

- New set up and existing set up: Check whether the broker is down.

- Existing set up: It's possible you did not configure all brokers that contain the topic the consumer connects to, and the brokers which are configured in that consumer are down.

- New set up: If you are encountering SSL connection-related errors, check the steps that you used to import certificates to both EB and consumers.

An EB component crashes: ArcMC Rest API, stream processors (Routing and Transform)

- At Start Up: Check the container start up order (above). Have any of the dependency pods not started or crashed?

- Memory: JVMs require more memory that the system has available.

- All: check whether there are too many open sockets.

EB EPS is lower than expected

- Check whether there are resource constraints on brokers: CPU, memory, disk is full. Check usage at system level (or with ArcMC).

- Network bottleneck.

- Stream processor is not able to keep up with transformation; SP is constrained in some way (resources). In ArcMC, the Stream Processor metric will be lower than the connector EPS.

  Check that you have sufficient resources

- Expected file system

- Memory

- CPU

- File descriptor loads

FAQs

Which pods in Kubernetes comprise the Event Broker deployment?

- Event broker pods
    - confluentinc pods: kafka; schemaregistry; zookeeper
    - EB pods: c2av-processor, kafka-manager; orches; routing-processor; web-service

Related topic: ArcSight Event Broker prerequisites

Which pods in Kubernetes comprise the ArcSight Event Broker deployment?

- Hercules pods: management, proxy, rethinkdb, search

Related topic: ArcSight Event Broker prerequisites

How can I find out more about Kafka and Apache ZooKeeper?

See these resources for more about Kafka and Apache ZooKeeper.

- Kafka: https://sookocheff.com/post/kafka/kafka-in-a-nutshell
- Benchmarking Apache Kafka: 2 Million Writes Per Second (On Three Cheap Machines) | LinkedIn Engineering
- How to choose the number of topics/partitions in a Kafka cluster? - Confluent
- Apache Kafka
- Introduction to Kafka and ZooKeeper
- Introduction to Apache ZooKeeper | Apache ZooKeeper Tutorials Setting up Apache ZooKeeper Cluster | Apache ZooKeeper Tutorials

Can I use my existing Event Broker v1.0 with ArcSight Investigate + Vertica?

No. ArcSight Investigate requires Event Broker 2.0. You can migrate your data from Event Broker 1.0 using the Event Broker Data Migration utility. Check with ArcSight Support about the availability of this tool.

Related topic: Investigate 1.0 Deployment Guide. See also the Event Broker Data Migration Tech Note when available.

With only a single install method for Event Broker, how can I distribute each of these (pods) across Kubernetes worker nodes?

The ArcSight Installer application automatically manages the distribution and deployment of Kubernetes worker nodes.

Related topic: Adding Kubernetes nodes

How do I check the hostname of machines on which EB is installed?

Example:

```
[root@n11.222.444.h11 ~]# kubectl get node  -L fdqn

NAME             STATUS    AGE       FDQN

11.222.333.222   Ready     1d        n11.222.333.h222.domainname.com

11.222.444.11    Ready     1d        n11.222.444.h11.domainname.com

After checking labels on the above EB, it shows both master and work
node are using the same label. Need to make sure they are correct Label.
```

To verify whether the hostname is set correctly:

```
# ssh root@11.222.777.111  "hostname -f"

root@11.222.777.111's password:

n11.222.777.h111.domainname.com

# ssh root@11.222.777.111  "nslookup 11.222.777.111 | grep 'name ='"

root@11.222.777.111's password:

111.333.222.11.in-addr.arpa     name = n11.222.777.n111.domainname.com.
```