Hewlett Packard Enterprise

HPE Security ArcSight ADP Event Broker

ソフトウェアバージョン: 2.02

展開ガイド

2017年8月15日

ご注意

保証

Hewlett Packard Enterprise製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、Hewlett Packard Enterprise Development LPはいかなる責任も負いません。ここに記載する情報は、予告なしに変更されることがあります。本書の例で使用しているネットワーク情報 (IPアドレスやホスト名を含む) は、説明のみを目的としています。Hewlett Packard Enterprise Development LPの製品は高い柔軟性を持ち、お客様の設定に応じて機能します。データのアクセス性、完全性、機密性については、ユーザーが責任を負います。包括的なセキュリティ戦略を実施し、優れたセキュリティ慣習に従ってください。 本書は機密情報です。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、Hewlett Packard Enterprise Development LPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

サポート

連絡窓口

電話	電話番号の一覧は、HPE Security ArcSightテクニカルサポートページに記載されています。 https://softwaresupport.hpe.com/documents/10180/14684/esp-support-contact-list
サポートWebサイト	https://softwaresupport.hpe.com
Protect 724コミュニティ	https://www.protect724.hpe.com

目次

ArcSight Event Brokerについて 5

サポートされる展開シナリオ 7

ArcSight Event Brokerの展開アーキテクチャー 7

Kubernetesノードの管理方法について 9

ArcSight Event Brokerの展開の概要 11

次の内容 11

ArcSight Event Brokerの前提条件 12

システム要件 12

プロデューサー/コンシューマーのインターフェイスと暗号化モードの準備 13

Kubernetesで管理するシステムの準備 15

TLSの計画 15

すべてのシステムの準備 16

インストール順序 17

次の内容 17

ArcSightインストーラーアプリケーションを 使用したArcSight Event Brokerのインストール 18

マスターノードでのワーカーノード用のキーペアの生成 18

マスターノードでのArcSightインストーラーのインストール 19

Kubernetesのインストールとセットアップ 20

展開前のニーズに合わせたinstaller.propertiesの調整 23

オフラインダウンロードとローカルDocker Hubの セットアップ手順 (展開前) 26

ArcSightインストーラーでのEvent Brokerワーカー ノードの展開 28

ArcSight Event Brokerのコンポーネントの設定 29

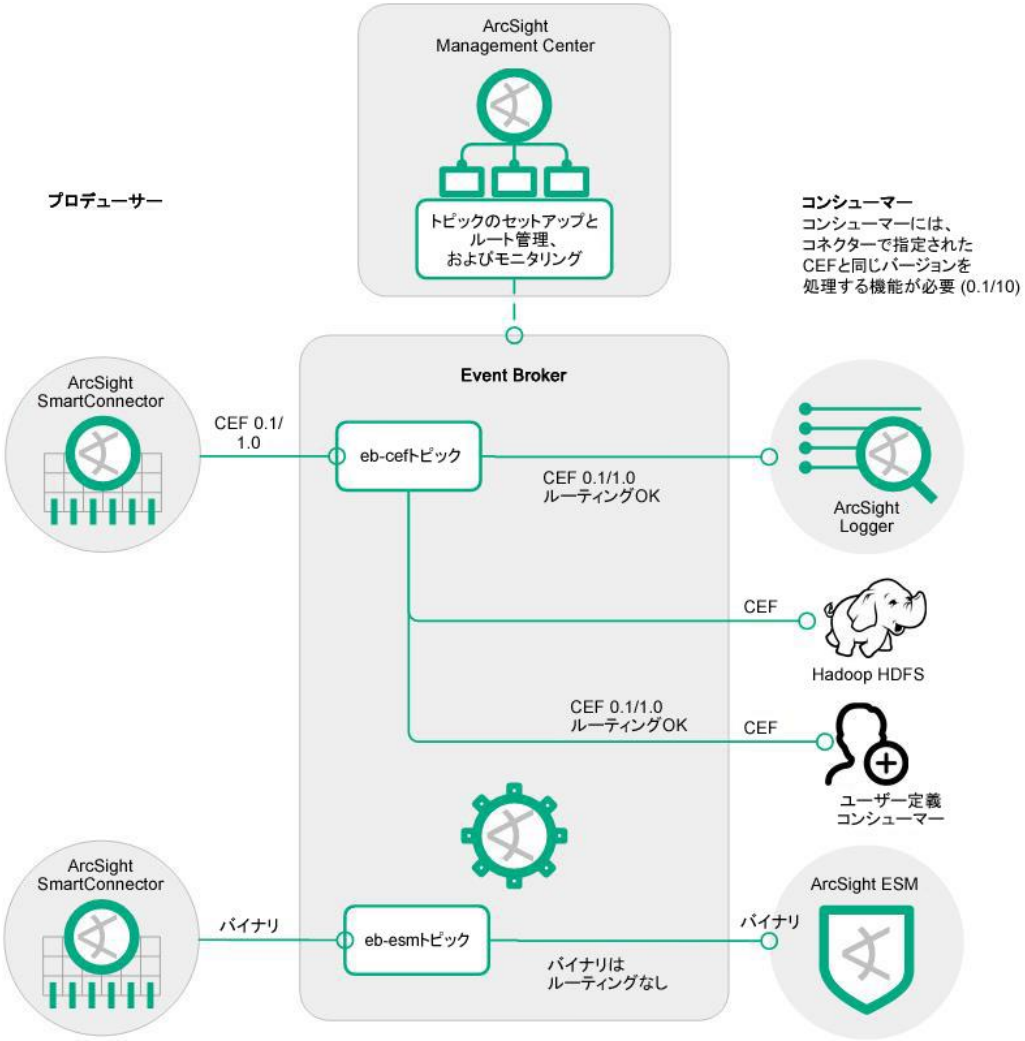
Event BrokerのArcMCによる管理の設定 29

SmartConnectorの設定 29

Kubernetesノードの追加 30
コンシューマー用の署名済み証明書の生成 30
システムCAからの署名済み証明書の生成 31
CSRからの署名済み証明書の生成 31
ArcSight Event Brokerのアンインストール 32
Event Brokerのアップグレード 33
オフラインアップグレード 34
次のステップ 35
ArcSight Event Brokerの展開の トラブルシューティングとFAQ 36
トラブルシューティング 36
FAQ 40

ArcSight Event Brokerについて

ArcSight Event Brokerは、トピックのソートやイベントのルーティングにより、イベントを一元処理するソリューションです。ArcSight環境を拡張し、ArcSightイベントデータをサードパーティ製ソリューションから利用できるようにします。優れた拡張性と可用性を備えたマルチブローカークラスターを実装することで、イベントデータのパブリッシュとサブスクライブを行います。ADP Event Brokerは、ArcSightコネクター、Logger、ESMと統合されます。これによってArcMCによる管理と監視が可能になり、ArcSight Investigateを使用する基盤になります。ArcSight Data PlatformのEvent Brokerは、Apache Kafkaをパッケージ化したものです。Event BrokerのKafkaブローカーまたはブローカーノードのクラスターをインストールして設定すると、ADP SmartConnectorを使用してデータをパブリッシュし、ADP Logger、ArcSight Investigate、Apache Hadoop、または独自のコンシューマーでそのデータをサブスクライブできます。



コンポーネント	説明
ArcSight インストーラー	ArcSight InvestigateやEvent Brokerなどの、ArcSightコンポーネントの展開と設定を行うためのWebアプリケーション。 これらのコンポーネントは、Kubernetesクラスターで管理されます。
ArcSight SmartConnector	SmartConnectorは、ネットワーク上のデバイスからイベントデータを収集し正規化します。コネクターでイベントデータを正規化する方法には、緊急度、重要度、タイムゾーンなどの値を共通フォーマットに正規化する方法と、データ構造を共通スキーマに正規化する方法の2通りの方法があります。 SmartConnectorは続いてイベントのフィルタリングとアグリゲーションを行うことで、システムに送信されるイベントのボリュームを削減できます。ArcSight SmartConnectorは、個別にインストールおよび管理され、Event Brokerにデータをパブリッシュするプロデューサーとして機能します。Event Brokerによって管理されるデータは、ArcSight Investigate、ArcSight ESM、ADP Logger、Apache HDFS、またはサードパーティ製コンシューマーでサブスクライブできます。
Event Broker	ArcSight Event Brokerはイベント処理を一元化します。これにより、イベントデータのパブリッシュとサブスクライブに対応した、高スループットでスケーラブルなマルチブローカークラスターの利用が可能になります。Event Brokerでデータストリームの調整と管理を行うことで、お使いのArcSight環境を拡張し、ArcSightイベントをサードパーティ製データソリューションで利用することが可能になります。
ArcMC	HPE ArcSight Management Center (ArcMC) は、セキュリティポリシーの設定、展開のメンテナンス、および監視を、効率的かつ低コストで簡単に実行できる一元管理ツールです。ArcMCにより、Event Brokerトピックの実行時管理が可能になります。ArcMCは、ArcSight Deployment Platform (ADP) の一部として販売されています。

Event Brokerは、コンシューマーがサブスクライブできるトピックのイベントの分配を管理します。

- プロデューサー側のコネクターで設定するCEFバージョン (CEF 0.1または1.0) は、コンシューマーがサポートしているCEFバージョンである必要があります。
- 2種類のEBトピック (eb-cefおよびeb-esm) を、SmartConnectorの接続先として設定できます。ESMトピックでは、ESMで使用する形式であるバイナリが生成されます。
- 複数のコネクターを、eb-cefトピックにパブリッシュするように設定できます。負荷分散はEBで行われます。
- Event Brokerのストリームプロセッサーは、eb-cefトピックのCEF形式のイベントデータをAvro形式に変換し、Avro形式のイベントデータをeb-internal-avroトピックに送信します。
- ArcSight ESMは、Event Brokerのコンシューマー (eb-esmトピックからのデータを使用) として設定できます。

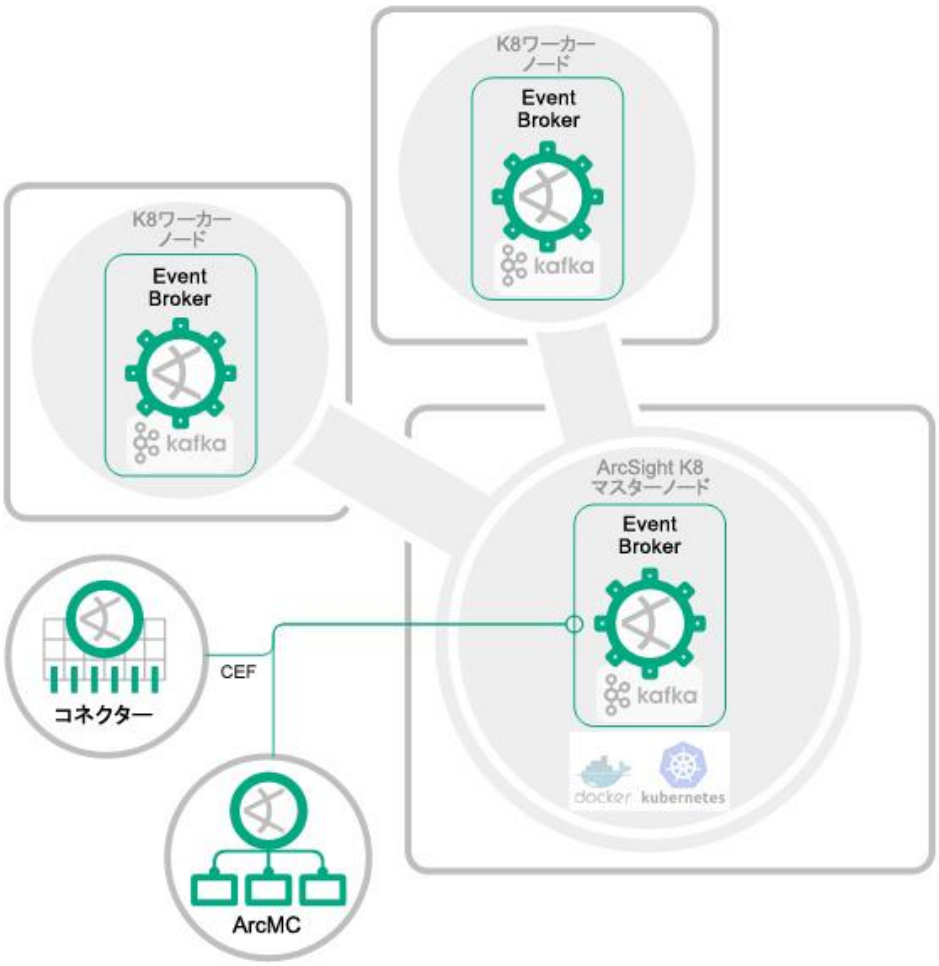
サポートされる展開シナリオ

ArcSight Event Brokerは、次の展開シナリオをサポートしています。

	ユースケース	説明	ガイド文書
1	EB 1.0を使用しているADP 2.0の既存のお客様がArcSight Investigateを追加する	1. EB 1.0を2.0にアップグレードし、EB 1.0データをEB 2.0に移行します。データ移行手順の利用可否について、カスタマーサポートに問い合わせます。 2. ArcMCを2.6または2.6.1にアップグレードします。 3. ArcSightインストーラーアプリケーションから、Investigateをインストールします。	1. EB data migration tech note 2. ArcSight Management Center管理者ガイド 3. Investigate Deployment Guide
2	スタンドアロンのEB 2.0 (Investigateなし)	ArcSightインストーラーアプリケーションのスタンドアロンEBバージョンを使用して、新規インストールを行います。EBはArcMC 2.6以降で管理します。	Event Broker管理者ガイド ArcSight Management Center管理者ガイド
3	スタンドアロンEB 1.0からEB 2.0へのデータ移行	EB 1.0からEB 2.0へのデータの移行手順	EB data migration tech note
4	EB 2.01からEB 2.02へのアップグレード	Event Brokerの既存のインストール環境を、Event Broker 2.02にアップグレードします。	Event Broker展開ガイド

ArcSight Event Brokerの展開アーキテクチャー

ArcSight Event Brokerは、Kubernetesで管理するDockerコンテナーを使用してインストールし、ArcSightインストーラーアプリケーションから展開します。デフォルトの展開構成は、1つのKubernetesマスターノードと2つのKubernetes (k8s) ワーカーノードから成ります。



展開コンポーネント	ホスト	機能内容
Kubernetes マスターノード	1つのVMまたは物理サーバー	• Kubernetesマスターノード • Kafka Event Brokerノード
Kubernetes ワーカーノード	2つのVMまたは物理サーバー	• 2つのKafka Event Brokerノード
ArcSight SmartConnector	スタンドアロンまたはコネクターをホストしているアプライアンス上にホスト	ネットワークデバイスからのイベントデータを正規化し、CEF形式にします。

展開コンポーネント	ホスト	機能内容
ArcSight管理コンソール	別にインストール	Event Brokerのトピックの実行時管理を行います。

Kubernetesノードの管理方法について

ArcSight Event Brokerは、エラスティックスケーリングを可能にするKubernetesコンテナー管理を使用して、ArcSightインストーラーアプリケーションによってインストールおよび展開されます。Kubernetesマスターノードコントローラーは、1つのシステム/ノード上に配置されます。Kubernetesワーカーノードには、Podと呼ばれるコンテナー管理ユニットがホストされます。Podでは、共有名前空間と共有ボリュームを使用して、1つまたは複数のコンテナーを管理します。

デフォルトシステムPod	Event BrokerのPodおよびコンテナー	ArcSight InvestigateのPodおよびコンテナー
default-http-backend-*	eb-c2av-processor-* (Pod) c2av-processor (コンテナー)	hercules-management-* (Pod) kubernetes-vault-renew (コンテナー) hercules-management (コンテナー)
nginx-ingress-controller-*	eb-kafka-0 (Pod) kafka (コンテナー)	hercules-rethinkdb-0 rethinkdb (コンテナー)
	eb-kafka-1 (Pod) kafka (コンテナー)	hercules-search-* (Pod) kubernetes-vault-renew (コンテナー) hercules-serach-engine (コンテナー) hercules-search (コンテナー)
	eb-kafka-2 (Pod) kafka (コンテナー)	

デフォルトシステムPod	Event BrokerのPod およびコンテナー	ArcSight InvestigateのPodおよびコンテナー
	eb-kafka-manager-* (Pod) kafka-manager (コンテナー)	
	eb-routing-processor-* routing-processor (コンテナー)	
	eb-schemaregistry-* (Pod) schemaregistry (コンテナー)	
	eb-web-service-* (Pod) kubernetes-vault-renew (コンテナー) atlas-web-service (コンテナー)	
	eb-zookeeper-0 (Pod) zookeeper (コンテナー)	
	eb-zookeeper-1 (Pod) zookeeper (コンテナー)	
	eb-zookeeper-2 (Pod) zookeeper (コンテナー)	

セットアップおよび展開時に、アスタリスク (*) 付きのPodには、ランダムに生成される識別番号が付加されます。

Kafkaワーカーノードは、冗長性と負荷分散を実現します。あるノードが機能しなくなるかオフラインになると、他のノードに引き継がれます。

KafkaまたはZooKeeperでは、デフォルト構成にノードを追加できます。ノードを追加する際には、installer.propertiesファイルを変更し、追加のリソースプランニングを行う必要があります。詳細については、「Kubernetesノードの追加」を参照してください。

ArcSight Event Brokerの展開の概要

1. ArcSight Event Brokerの前提条件
2. ArcSightインストーラーアプリケーションを使用したArcSight Event Brokerのインストール
3. ArcSight Event Brokerのコンポーネントの設定
4. Kubernetesノードの追加
5. ArcSight Event Brokerのアンインストール
6. ArcSight Event Brokerの展開のトラブルシューティングとFAQ

次の内容

ArcSight Event Brokerの前提条件

ArcSight Event Brokerの前提条件

システム要件

ここでは、デフォルトのセットアップに基づいた一般的なサイズ要件について説明します。実稼働環境に応じた推奨サイズについては、ArcSightカスタマーサポートまでお問い合わせください。

展開に使用する以下の要件に対応したサーバー (またはVM) をプロビジョニングします。サポートされるプラットフォームとオペレーティングシステムについては、Protect 724の『ADP Support Matrix』ドキュメントを参照してください。

コンポーネント	ノード数	必要なリソース	使用可能なポート
Event Broker	1 (マスター) 2 (ワーカーノード)	• 1 CPU (24コア) • 32 GB RAM • 8 TBのディスク容量 • RHELまたはCentOS 7.3 • Linuxカーネルバージョン3.10以上 • 10 GigEネットワーク (またはこれと同等のもの)	32181、9093、9092
ArcMC (ADPの一部)	1	• 1 CPU (クアッドコア) • 16 GB RAM • 50 GBの空きディスク容量 ArcMCの展開の詳細については、『ArcSight Management Center管理者ガイド』を参照してください。	
SmartConnector (ADPの一部)	1	SmartConnectorバージョン7.6 (スタンドアロンまたはArcMCにより管理) ArcSight SmartConnectorの展開の詳細については、『SmartConnectorユーザーガイド』を参照してください。	

プロデューサー/コンシューマーのインターフェイスと暗号化モードの準備

オプションのプロデューサー/コンシューマーコンポーネントを準備する際には、以下のガイドラインに従ってください。詳しい手順については、指定されたガイド文書を参照してください。ArcSight Event Brokerの展開後のプロデューサーおよびコンシューマーの設定方法については、「ArcSight Event Brokerのコンポーネントの設定」を参照してください。

Event Brokerのインストールおよび設定前の暗号化モードのセットアップ

Event Brokerをインストールする前に、ArcSightコンポーネント間の通信を暗号化するのに使用する暗号化モードを決めます。他のArcSightコンポーネントは、Event Brokerに接続する前に、使用する暗号化モードを用いてセットアップします。Event Brokerに接続するシステム (コンシューマー、プロデューサー、ArcMC) のセキュリティモードは、Event Brokerで設定するセキュリティモードと同じなければなりません。Event Brokerの展開後に暗号化モードを変更する場合は、システムのダウンタイムが必要になります。Event Brokerの展開後にセキュリティモードを変更する必要がある場合は、『Event Broker管理者ガイド』を参照してください。

製品	**必要な準備**	**オープンポート**	**サポートされる暗号化モード**	**ガイド文書**
ArcMC	ArcMCは、ArcSight Event Brokerをインストールする前にインストールします。	38080	• TLS • FIPS • ClientAuth	ArcSight Management Center管理者ガイド

製品	必要な準備	オープンポート	サポートされる暗号化モード	ガイド文書
ArcSight SmartConnector	ArcSight SmartConnectorとArcMCオンボードコネクターは、ArcSight Event Brokerをインストールする前にインストールして実行できます。 コネクターバージョン7.6とEvent Brokerとの間では、FIPSモード設定はサポートされません。SmartConnectorバージョン7.6とEvent Brokerとの間でサポートされる暗号化方式は、TLSとClientAuthのみです。	9093	• TLS • FIPS • ClientAuth	SmartConnectorユーザーガイド ArcSight Management Center管理者ガイド
ArcSight ESM (オプション)	ArcSight ESMは、ArcSight Event Brokerをインストールする前にインストールして実行できます。	9093	• TLS • FIPS • ClientAuth	ESMインストールガイド ESM Administrator's Guide
ArcSight Logger (オプション)	ArcSight Loggerは、ArcSight Event Brokerをインストールする前にインストールして実行できます。	9093	• TLS • FIPS • ClientAuth	Logger管理者ガイド

Kubernetesで管理するシステムの準備

Kubernetesで管理するすべてのシステムについて、以下を実行します。

1. 上記のガイドラインを用いて、Kubernetesのマスターノードとワーカーノードを準備します。
2. ノード間の通信用に、公開キー暗号が必要です。
    a. キーペアを生成するには、マスターノードで以下のコマンドを実行します。
    ssh-keygen -t rsa
    b. マスターノードで以下のコマンドを実行して、公開キーをワーカーノードにコピーします。
    ssh-copy-id -i ~/.ssh/id_rsa.pub root@<workernode_hostname>
3. クラスター内のすべてのホストで、Chronyを使用してNTPを設定します。
    Chronyは、Red HatおよびCentOSの一部のバージョンではデフォルトでインストールされています。
    Chronyがインストールされていない場合は、以下のコマンドを実行します。
    a. Chronyをインストールします。

    ```
    yum install chrony
    ```

    b. Chronydを開始します。

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

    c. Chronyが正しく動作していることを確認します。

    ```
    chronyc tracking
    ```

TLSの計画

ArcSight Event Brokerシステム内の各種コンポーネントは、TLS 1.2プロトコルを使用して実装された暗号化通信を用いてやり取りします。

TLSの実装には、デジタル証明書が必要です。インストールを始める前に、使用する証明書のタイプを決める必要があります。

- Kubernetesの自己署名証明書: Kubernetesには、自己署名証明書を生成する機能があります。デフォルトでは、Kubernetesのインストールプロセスにより、Kubernetesクラスター用の証明書が生成されますが、インストール中にそれ以外の方法を指定することもできます。また、ArcSight InvestigateのVerticaデータベースなど、証明書を必要とするシステム内の他のコンポーネント用にKubernetes証明書を生成することもできます。Kubernetes証明書の生成の詳細については、システムCAからの署名済み証明書の生成を参照してください。

- CA (認証局) によって署名された有効なデジタル証明書: 組織のセキュリティポリシーによっては、信頼済みのCAからの証明書が必要になる場合があります。この場合は、ルート証明書ファイルと秘密キーファイルが必要です。これらのファイルを指定されたKubernetesマスターノードにコピーします。

注: インストール後に証明書を再設定することはできません。

FIPSモードを有効にすることを計画している場合は、生成した証明書がFIPSの条件を満たすようにする必要があります。

すべてのシステムの準備

すべてのサーバー、マスターノード、ワーカーノードで、以下を実行します。コマンドはすべて、rootユーザーとして実行する必要があります。

- ファイアウォールを有効にします (ArcSightインストーラーによって必要なポートが開かれます)。`systemctl enable firewalld`コマンドと`systemctl start firewalld`コマンドを実行します。
- 製品のコンテナーイメージをダウンロードするため、インターネットアクセスを有効にします。
- 組織のネットワークでプロキシが使用されている場合は、すべてのサーバーでプロキシ環境変数を定義します。
- マスターノードとワーカーノードをすべてリブートします。
- 各ノードが完全修飾ドメイン名を使用して設定されていることを確認します。
- フォワードプロキシやリバースプロキシのルックアップを含めて、すべてのシステムでDNSが正しく設定されていることを確認します。
- SELinuxを無効にします。
- 次のようにして、デフォルトのユーザープロセス (ulimit) を増やします。

    a. /etc/security/limits.d/*-nproc.confファイルを開きます。
        1. /etc/security/limits.d/*-nproc.confファイルがない場合は、作成します (必要な場合は、limits.dディレクトリも作成します)。
        2. このファイルがすでに存在する場合は、ファイル内のすべてのエントリを削除します。
    b. 以下の行を追加します。新しいエントリには、必ずアスタリスク (*) を含めてください。すべてのエントリを指定どおり正確に追加することが重要です。

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

プロキシサーバーのセットアップ

環境内でプロキシサーバーを使用している場合は、すべてのサーバーでプロキシ環境変数を定義します。

手順

1. CentOSのリポジトリにyumでアクセスできるように、(必要に応じて) プロキシ設定を構成します。

```
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. Kubernetesマスターノードとワーカーノードをすべてリブートします。ノードはどの順序でリブートしても構いません。

インストール順序

ArcMCとArcSight SmartConnectorは、ArcSight Event Brokerの展開時に任意の順序でインストールできます。ただし、ArcSightでは、Event Brokerのコンポーネントをインストールする前に、これらを環境内にインストールして実行しておくことを推奨します。ArcSight Event Brokerをインストールした後で、両方のコンポーネントの追加設定が必要になります。

次の内容

ArcSightインストーラーアプリケーションを使用したArcSight Event Brokerのインストール

ArcSightインストーラーアプリケーションを使用したArcSight Event Brokerのインストール

ArcSightインストーラーアプリケーションを実行する前に、Event Brokerの前提条件のガイドラインに従って、受信システムのセットアップが完了していることを確認します。ArcSightインストーラーは、Kubernetesのマスターノードとワーカーノードの両方で、セットアップ時にファイアウォールの設定を行います (firewalld.serviceが稼働している場合)。マルチマスター型のインストールはサポートされません。

ここでは、Docker Hubリポジトリを使用してオンラインインストールを行う手順と、FTPサイトからtarファイルをダウンロードし、マスターノードシステムにローカルDocker Hubを複製することでオフラインインストールを行う手順について説明します。

マスターノードでのワーカーノード用のキーペアの生成

マスター/ワーカーノードの構成では、マスターノードでキーペアを生成し、公開キーを各ワーカーノードにコピーします。これにより、マスターサーバーからクラスター内の他のすべてのワーカーノードサーバーに対して、パスワードを使用せずにSSHアクセスできるようになります。この作業は、ArcSightインストーラーをインストールする前、およびKubernetesのインストールとセットアップを行う前に行います。

以下に、パスワードを使用しないSSHアクセスを有効にする手順の例を示します (オンライン上には、http://www.linuxproblem.org/art_9.htmlのような異なる手順の例もあります)。

注: キーペアの生成はrootユーザーで行います。

手順

1. マスターサーバーでssh-keygenコマンドを実行します。以下に例を示します。

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

2. キーを保存するファイルを入力します。以下に例を示します。

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

3. パスフレーズを入力します。もう一度、パスフレーズを入力します。
4. ワーカーノードのIPアドレスを使用して、マスターノードからワーカーノードにキーをコピーします。以下に例を示します。

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

5. キーのフィンガープリントが表示され、ワーカーノードサーバーを認証するように求められます。必要に応じて、ワーカーノードのパスフレーズを入力します。次のメッセージが表示されたら、この操作は成功です。

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

6. ワーカーノードにキーが正常にインストールされたことを確認するには、マスターノードからワーカーノードに次のコマンドを実行して、ワーカーノードに正常にログインできることを確認します。

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

7. すべてのワーカーノードに対して、ステップ4～6を繰り返します。

マスターノードでのArcSightインストーラーのインストール

以下の手順では、ArcSightインストーラーをインストールし、アプリケーションを起動して、Kubernetesのスクリプトを/opt/arcsight/installer/k8sにアンパックします。

注: インストールは、rootユーザーまたはsudoユーザーで行います。

手順

1. インストールパッケージとMD5チェックサムファイルをマスターサーバーにコピーします。
2. インストールパッケージが完全な状態であることを確認します。md5sum <installer filename>とcat <installer filename>を実行します。この2つのチェックサムは同じ値になるはずです。
3. マスターサーバーで、ディレクトリをインストールパッケージの場所に変更し、次のコマンドを実行します(例: "arcsight-installer-x.x.x.x86_64.rpm")。

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

Kubernetesのインストールとセットアップ

以下の手順では、マスターノード上にKubernetesを展開し、ワーカーノードをセットアップします。

注: インストールは、rootユーザーで行います。

手順

1. マスターサーバーで次のスクリプトを実行します。

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

オプションパラメーター

a. Dockerのログローテーションのデフォルト設定を更新するには、次のパラメーターを使用します (デフォルトで、Dockerのログローテーションでは、コンテナーごとに合計5つのファイルを保持し、各ファイルの最大サイズは20 MBです)。

LOG_MAX_SIZE: Dockerのログローテーションでの、コンテナーごとの最大ログファイルサイズを指定します。数値に続けて単位を指定します (k=キロバイト、m=メガバイト、g=ギガバイト)。

LOG_MAX_FILE: Dockerのログローテーションで、コンテナーごとに保持する最大ファイル数を指定します。
LOG_MAX_SIZEおよびLOG_MAX_FILEの例:

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

b. 信頼済みCAからの証明書を使用する場合は、次のパラメーターを使用します。

ROOTCA: クライアントおよびサーバー証明書を生成するためのルートまたは中間証明書。

ROOTCAKEY: クライアントおよびサーバー証明書を生成するための秘密キー。

CLOUD_PROVIDER: このパラメーターは、クラウドプロバイダーがMicrosoft Azure (専用の設定が必要) である場合に必要です。値として、azureを渡します。

ROOTCAおよびROOTCAKEYの例:

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

CLOUD_PROVIDERの例

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

注: プロキシを使用する必要があり、オフラインインストールを行っていない場合は、この次のステップで、空のプロンプトに次の形式でプロキシを入力します。`http://proxy.example.com:80/` (空のプロンプトは見過ごしがちです。)

オフラインインストールを行っている場合は、空のプロンプトで、Enterキーを押して空のプロンプトをスキップします。このステップで情報を入力する必要はありません。

2. マスターサーバーで、ワーカーノードごとに以下のコマンドを実行します。ワーカーノードのIPは、IPv4形式 (1.1.1.1) で指定します。

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

3. ArcSightインストーラーが正常にインストールされたことを確認します。マスターサーバー上で、ArcSightインストーラーのURLをブラウザーで開きます。ブラウザーにArcSightインストーラーのログイン画面がロードされます。自己署名証明書を使用する場合は、ブラウザーからの自己署名証明書通知を受け入れます。

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

4. 新しいクラスターを作成します。

    a. マスターサーバー上でArcSightインストーラーを開きます。

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

    b. ログインページで、[**Create New Cluster**] をクリックします。
    c. [**Add New Cluster**] ページで、以下の情報を入力し、[**Create**] をクリックします。

        - **Cluster ID**: 英字数字とアンダースコアで構成されるIDを入力します。5文字以上である必要があります。スペースはサポートされていません。
        - **Password/Confirm Password**: 6文字以上のパスワードを入力し、確認のため再入力します。

5. kubectl labelコマンドを使用して、すべてのノードにラベルを付けます (以下に示すコマンドでは、マスターノード1つとワーカーノード2つの3ノード構成にラベルを付けています。必要に応じて、それぞれの構成に基づいて、クラスター内のすべてのノード`<workerN_ip>`にラベルを適用します)。

```
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. KubernetesマスターノードおよびDocker Hubへの接続をセットアップします。
    a. ArcSightインストーラーの [**Cluster Setup**] ページで、[**Master**] フィールドの横の [**Edit**] をクリックします。
    b. [**Master Configuration**] ページで、以下の情報を入力し、[**Save**] をクリックします。
        - **MasterAddress**: マスターサーバーのホスト名またはIPアドレスを入力します。
        - **Token**: マスターノードの以下の場所から、トークンの最初のカンマ (,) より前にある内容 (XXXX,admin,adminのXXXXなど) を、このフィールドにコピーします。
          /opt/arcsight/kubernetes/ssl/token
        - **CA Certificate**: マスターノードの以下の場所から、CA証明書の内容をこのフィールドにコピーします。
          /opt/arcsight/kubernetes/ssl/ca.crt
7. Docker Hubを設定します。[Cluster setup] ページ > [Docker Repository] フィールドで、[**Edit**] をクリックして、Docker Hubを設定します。
    a. オフラインインストールを行っている場合は、オフラインインストール手順のセクションを参照してください。URLには、127.0.0.1:5000を入力し、UserName、Password、Emailにはダミーデータを使用します。Emailは有効なメール形式にしてください。
    b. オンラインインストールを行っている場合は、ダイアログにDocker HubのURL (index.docker.io) を入力し、それぞれのアカウントのUserName、Password、Emailを使用します。
8. Kubernetesマスターノードが正常にインストールされたことを確認します。
    a. ArcSightインストーラーで、[**Node Management**] をクリックし、インストールしたサーバーが**READY**になっていることを確認します。

注: [Node Management] ページで設定したノードが**READY**と表示されるまでに数分かかります。

数分たってもノードがREADYにならない場合は、ページを更新してください。

.

ステータス	説明
NOT_READY	ノードは使用可能ですが、製品を展開する準備ができていません。
READY	ノードは使用可能で、製品を展開する準備ができています。
ERROR	ノードは使用可能ですが、システムエラーにより展開できません。エラーアイコンにマウスカーソルを合わせると、メッセージが表示されます。

展開前のニーズに合わせたinstaller.propertiesの調整

ノードを展開する前に、それぞれの環境のニーズに合わせてプロパティを調整します。FIPSモードで展開する場合や、デフォルト構成にワーカーノードを追加する場合は、ここで設定したプロパティの調整が必要になります。

プロパティファイルの設定	デフォルト値	オプション
Event Brokerのすべてのコンポーネントで FIPS認定暗号アルゴリズムを使用	predeploy.eb.init.fips=false	FIPS暗号をセットアップする場合は、trueに設定します。
Event BrokerのKafkaでクライアント接続を確認するのにTLSクライアント認証を使用	predeploy.eb.init.client-auth=false	クライアント認証をセットアップする場合は、trueに設定します。

プロパティファイルの設定	デフォルト値	オプション
KafkaのEvent Brokerトピック用パーティション数	predeploy.eb.init.noOfTopicPartitions=6	必要に応じて、異なる数のパーティションを設定します。
KafkaのEvent Brokerトピック用レプリケーション係数	predeploy.eb.init.topicReplicationFactor=2	必要に応じて、Kafkaに異なるレプリケーション係数を設定します。
Kafkaのログ保持サイズ (バイト)	predeploy.eb.init.kafkaRetentionBytes=10737418240	必要に応じて、Kafkaのログ保持サイズに異なるサイズ (バイト単位) を設定します。
KafkaのVertica avroトピック用のログ保持サイズ。これは非圧縮であるため、同じ期間のイベントを保持するのにより大きな領域が必要です (バイト)	predeploy.eb.init.kafkaRetentionBytesForVertica=10737418240	必要に応じて、Vertica Avroトピックに異なるサイズ (バイト単位) を設定します (存在する場合)。
Kafkaのログ保持期間 (時間)	predeploy.eb.init.kafkaRetentionHours=672	必要に応じて、Kafkaのログ保持に異なる期間を設定します。

プロパティファイルの設定	デフォルト値	オプション
Kafkaのブローカー間プロトコルのバージョン	predeploy.inter.broker.protocol.version=0.10.1.0	アップグレード時にのみ変更されます。
ブローカーがログにメッセージを追加するのに使用するメッセージ形式バージョン	predeploy.log.message.format.version=0.10.1.0	アップグレード時にのみ変更されます。
KafkaおよびZooKeeperノードの数	predeploy.eb.kafka.count=3 predeploy.eb.zookeeper.count=3	
データを永続的に保存するためのホストパス	predeploy.eb.kafka.path=/opt/arcsight/k8s-hostpath-volume/eb/kafka predeploy.eb.zookeeper.path=/opt/arcsight/k8s-hostpath-volume/eb/zookeeper	
ArcMCホスト名	predeploy.eb.arcmc.hosts=localhost:443	管理元のArcMCの場所を指定します。
サーバー証明書を使用してサーバーのホスト名を検証するためのエンドポイント識別アルゴリズム	predeploy.ssl.endpoint.identification.algorithm=	
ストリームスレッドの数	predeploy.stream.num.threads=6	

プロパティファイルの設定	デフォルト値	オプション
C2avでのフィールドの切り詰め	predeploy.c2av.field.truncate=false	EB＋ArcSight Investigateの場合。イベントフィールドの情報サイズが定義された長さを超えた場合に、イベントがVerticaの拒否テーブルに送信されるのを回避するには、trueに変更します。
各EBコンテナーのログレベル	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}	デバッグを行う場合は、「info」を「debug」に変更します。
ArcMCの証明書のホストパスディレクトリ	predeploy.arcmc.certs.path=/opt/arcsight/k8s-hostpath-volume/eb/arcmccerts	この設定内容は、ArcMCでのEvent Brokerのセットアップ時に入力されます。

オフラインダウンロードとローカルDocker Hubのセットアップ手順 (展開前)

この手順は、ArcSight Event BrokerサーバーからHPE Docker Hubレジストリへのインターネット接続がない場合に利用します。

手順

1. インターネットに接続されたサーバーから、HPE Software Entitlementsポータル (http://www.hpe.com/software/entitlements) に接続し、オフラインインストーラーファイルをダウンロードします。

コンポーネント	オフラインインストーラーファイル名
ArcSightインストーラーアプリケーション	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のみ)	arcsight-installer-1.10.7-ga110_eb.x86_64.rpm

2. これらのrpmファイルとtarファイルを、マスターノードの任意の場所にコピーします。
3. イメージをプライベートローカルレジストリにプッシュします。プライベートDockerレジストリは、127.0.0.1:5000で設定され実行されており、すべてのノードからアクセスできます。マスターノードで次のコマンドを実行します。

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

4. docker inspectコマンドを使用して、ファイルが完全な状態であることを確認します。Event Brokerには、6つのEvent Brokerイメージが含まれています。下記のコマンドで、.Idはイメージの「Id」フィールドを表します。これは、Dockerイメージの一意の識別子です。次のコマンドを実行して、実際の出力と以下の表示内容を比較します。実際の出力が以下と同じである必要があります。

```
# 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
```

5. ArcSightインストーラーの [**Cluster Setup**] ページで、[**Docker Repository**] の横の [**Edit**] ボタンをクリックします。
6. [**Docker Hub Configuration**] ダイアログボックスで [**URL**] フィールドに、次のURLを入力します。

```
127.0.0.1:5000
```

7. [**Save**] をクリックします。

ArcSightインストーラーでのEvent Brokerワーカーノードの展開

手順

1. ArcSightインストーラーで、[**Deployment**] タブをクリックします。
    a. Docker Hubと直接接続してオンラインインストールを行う場合は、ステップ2に進みます。
    b. ダウンロードしたrpm/tarファイルと127.0.0.1: 5000上のローカルDocker Hubを使用してオフラインインストールを行う場合は、上記の「オフラインダウンロードとローカルDocker Hubのセットアップ手順」に従って、ローカルDocker Hubリポジトリにイメージをプッシュしておく必要があります。
2. Event Brokerの横の [**Deploy**] をクリックします。
ステータスカラムで展開ステータスを確認できます。最初のステータスは**IN_PROGRESS**です。Docker Hubリポジトリとの接続速度によっては、展開に時間がかかる場合があります。製品が展開されると、ステータスが**DEPLOYED**に変わります。

ステータス	説明
NOT_READY	ArcSightインストーラーはKubernetesと通信できません。展開は実行できません。
OFF	ArcSightインストーラーはKubernetesと通信できます。製品はまだ展開されていません。
IN_PROGRESS	展開または展開解除が開始され、進行中です。このステータスは、Kubernetesで1つ以上のコンテナーが再開されたときにも表示されることがあります (たとえば、製品の設定を変更した場合)。
DEPLOYED	製品は正常に展開され、実行中です。
ERROR	1つ以上の製品のコンテナーが破損しています。エラーアイコンにマウスカーソルを合わせると、メッセージが表示されます。コンテナーがクラッシュし、その後Kubernetesによって修正または再開されると、このステータスが表示された後にDEPLOYEDに変わる場合があります。

ArcSight Event Brokerのコンポーネントの設定

Event BrokerのArcMCによる管理の設定

- EBホストをIPアドレスで識別できることを確認します。
- ArcMCにEBホストを登録します。
- ArcMCでノードを管理します。

詳細については、『ArcMC 2.6管理者ガイド』の「ホストの追加」を参照してください。

SmartConnectorの設定

- SmartConnectorの設定画面で、それぞれのトポロジに合わせてEvent Brokerの通知先を設定します。詳細については、『SmartConnectorユーザーガイド』を参照してください。

- イベントが着信し、検索でイベントが返されることを確認します。
- 必要に応じて、コネクターを追加します。

Kubernetesノードの追加

ワーカーノードの追加がサポートされています。新しいワーカーノードが追加されると、ZooKeeperやKafkaの場合と同様に、ラベルを使用して特定のPodを割り当てることができます。ここでは、ワーカーノードを追加してKafkaクラスターノードを拡張する方法について説明します。

1. 新しいKafkaノードを追加するには、新しいワーカーノードを追加してラベル付けし、レプリカ数を更新して、インストーラーのプロパティを更新します。たとえば、既存の3ノードKafkaクラスターに2つのノードを追加して、5ノードKafkaクラスターを作成するには、次の手順を実行します。

   /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. 今後の展開/展開解除でKafkaノード数が正しくなるように、新しいKafkaノード数に合わせてinstaller.propertiesを更新します。

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

   ...

   predeploy.eb.kafka.count=5

   ...

   $

コンシューマー用の署名済み証明書の生成

Event Brokerのコンシューマーがセキュアな通信を確立するには、Event Brokerからの署名済み証明書が必要です。

署名済み証明書を生成する方法は複数あります。それぞれ異なるユースケースで使用します。

- ArcSightインストーラーには、システムで設定されているCA (Kubernetesまたは別の信頼済みCA) から署名済み証明書を生成するためのユーティリティが含まれています。このユーティリティを使用すると、ArcSight InvestigateのVerticaデータベースなど、システム内の他のコンポーネント用の証明書を生成できます。

- ArcSightインストーラーには、CSR (証明書署名リクエスト) ファイルから署名済み証明書を生成するためのユーティリティが含まれています。このユーティリティを使用すると、ESMやLoggerなど、クライアント認証用の証明書を生成できます。

注: これらの手順は、Kubernetesをインストールした後でないと実行できません。

システムCAからの署名済み証明書の生成

手順:

1. Kubernetesマスターサーバーで、次のコマンドを実行します。

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

次の引数が必要です。

- $1: FQDN - 完全修飾ドメインサーバー名

コマンドを実行したディレクトリ内に、次のファイルが作成されます。

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

2. これらのファイルを、証明書を生成したサーバーにコピーします。

CSRからの署名済み証明書の生成

手順:

1. KubernetesマスターサーバーにCSRファイルをコピーします。

2. Kubernetesマスターサーバーで、次のコマンドを実行します。

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

次の引数が必要です。

- $1: CSRファイル (完全パス)
- $2: 作成するCRTの名前 (crt拡張子なし)

証明書と秘密キーが作成されます。

3. これらのファイルを、証明書を生成したサーバーにコピーします。

ArcSight Event Brokerのアンインストール

アンインストールするには、次の手順をこの順序で実行する必要があります。

1. Kubernetesをアンインストールします。
2. ArcSightインストーラーをアンインストールします。

手順

1. Kubernetesをアンインストールします。

   すべてのワーカーノードとマスターサーバーで次のコマンドを実行し、リブートします。
   ```
   sh /opt/arcsight/kubernetes/uninstall.sh
   ```

必要に応じて、マスターノードとすべてのワーカーノードでrm -rf /opt/arcsight/を実行し、Event Brokerによって作成されたデータをすべて削除します (これを行うと、イベントと設定ファイルもすべて削除されます)。

注: Event Brokerをシステムに再インストールする場合は、必ずデータファイルを削除してください。 データファイルを削除しないと、Event Brokerが古いデータファイルを使用して実行されるため、正しく機能しません。

2. ArcSightインストーラーをアンインストールします。マスターサーバーで、以下のコマンドを次の順序で実行します。

   `yum erase -y arcsight-installer.x86_64` (Arcsightインストーラーをアンインストール)

   rm -rf /opt/arcsight/installer (/optフォルダーの下に残っているログファイルとディレクトリをすべて削除)

Event Brokerのアップグレード

Event Brokerをバージョン2.1にアップグレードするには、次の手順を実行します。

1. ArcSightインストーラーサービスを停止します。

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

2. Kubernetesディレクトリに移動し、ca.keyとca.crtを安全な場所に保管します。

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

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

3. マスターノードとすべてのワーカーノードで、Kubernetesをアンインストールします。

```
# ../sh uninstall.sh

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

4. yumを使用してArcSightインストーラーを削除します。

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

5. ArcSightインストーラーの最新バージョンと、関連するMD5チェックサムファイルをダウンロードします。MD5チェックサムを確認し、ダウンロードしたファイルをインストールします。

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

6. ステップ2で保管したキーを使用して、最新のKubernetesファイルをインストールします。

```
# 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. ArcSightインストーラーのWebホームページに移動します。インストーラーで、トークンのみを更新し(証明書は同じ)、Dockerの資格情報を入力します。新しいトークンの内容を取得するには、`cat /opt/arcsight/kubernetes/ssl/token`を実行し、トークンの最初のカンマ (,) より前にある部分 (XXXX,admin,adminのXXXXなど) をコピーします。

8. Event Brokerと (必要に応じて) 他の更新済み製品を展開します。
9. 新しいバージョンのEvent Brokerの設定項目を設定します。

オフラインアップグレード

Event Brokerのオフラインアップグレードでは、以下の手順を実行します。

1. ArcSightインストーラーサービスを停止します。

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

2. Kubernetesディレクトリに移動し、ca.keyとca.crtを安全な場所に保管します。

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

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

3. マスターノードとすべてのワーカーノードで、Kubernetesをアンインストールします。

```
# ../sh uninstall.sh

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

4. yumを使用してArcSightインストーラーを削除します。

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

5. ArcSightインストーラーの最新バージョンと、最新のオフライン用.tarファイルをダウンロードしてインストールします。

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

6. ステップ2で保管したキーを使用して、最新のKubernetesファイルをインストールします。プロキシサーバーを求めるプロンプトが表示されたら、Enterキーを押します。

```
# 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. ArcSightインストーラーのWebホームページ (https://<home IP>:8888) に移動します。以前と同じ資格情報を使用してログインします。
8. インストーラーで、トークンのみを更新します (証明書は同じ)。新しいトークンの内容を取得するには、`cat /opt/arcsight/kubernetes/ssl/token`を実行し、カンマ (,) より前にある部分 (XXXX,admin,adminのXXXXなど) をコピーします。
9. 以下の情報を使用してログインします。

```
URL: 127.0.0.1:5000

Username: <as previous>

Password: <as previous>

Email: <as previous>
```

10. Event BrokerおよびInvestigate (必要な場合) の最新のイメージをプッシュします。

```
cd /opt/arcsight/installer/k8s/master/pushImages.sh

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

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

11. 新しいバージョンのEvent Brokerの設定項目を設定します。
12. ArcSightインストーラーのUIから、Event BrokerとInvestigateを展開します。

次のステップ

アップグレード後に、他のシステムで次のような追加手順が必要になる場合があります。

ArcMC: アップグレード前にArcMCでEvent Brokerを管理していた場合、新しいEvent Brokerの証明書をArcMCにインポートするまでは、ArcMCの監視サマリーにEvent Brokerのアップグレード後のバージョンに関する情報が正しく反映されません。また、Event Brokerのコンシューマーである管理対象ホスト (管理対象のLoggerなど) でも、新しい証明書を受け取る必要があります。

詳細については、『ArcSight Management Center管理者ガイド』の「ホストの証明書のダウンロードとインポート」を参照してください。

Logger: LoggerをEvent Brokerのコンシューマーとして設定している場合は、アップグレード後に、Loggerレシーバーを無効にしてから再度有効にします。

ArcSight Event Brokerの展開のトラブルシューティングとFAQ

トラブルシューティング

ログの場所

ログにアクセスするには、次のコマンドを使用します: `kubectl logs eb<podname>`

Podの開始順序

EBの展開後、Podは次の順序で開始する設定になっています。ダウンストリームPodは、依存関係が満たされた後に展開されます。

1. クラスター内のZooKeeper Podのクォーラムが稼働状態になる必要があります (3つのうち2つ、または5つのうち3つ)。ZooKeeperの総数は奇数である必要があります。
2. Kafka Podがすべて稼働状態になる必要があります。
3. Schema Registry Podが稼働状態になる必要があります。
4. Bootstrap Web Service、Kafka Manager。
5. 変換ストリームプロセッサー、ルーティングストリームプロセッサー。

初めて展開を行う際のPodの再起動

Event Brokerの展開を初めて行う場合、Event BrokerのすべてのPodに対して必要なすべてのDockerイメージをDocker Hubから初めてダウンロードしている間に、一部のPodで再起動が発生しますが (Schema Registry Podなど)、これは通常の動作です。Event Brokerの展開時に、Podは特定の順序で起動するように構成されているため、ダウンストリームのPodは依存関係が満たされるまで展開されず、一定の待機時間および再試行回数後に再起動されます。

Dockerイメージのダウンロードは、ネットワークのダウンロード速度と構成に依存します。

ZooKeeperに対してクエリを実行できない

症状: kubectl get podsコマンドを実行してPodのステータスを取得すると、ダウンストリームのPod (Podの開始順序を参照) が稼働状態を維持できず、ステータスとして「CrashLoop」タイプのエラーが報告される。

以下を確認します。

- ZooKeeper Podが稼働していることを確認します。

- ZooKeeper Podのステータスが保留の場合、ノードにラベルがない可能性があります (zk=yes)。kubectl get nodes -L=zkコマンドを実行して、ノードにラベルがあることを確認してください。
- installer.propertiesのpredeploy.eb.zookeeper.count属性で、ZooKeeperの数が奇数に設定されていることを確認します。
- kubectl logs <pod name>を使用して、ZooKeeper Podログにエラーが記録されていないか確認します。

ZooKeeperログでよく報告されるエラーと警告

- クォーラム例外: リーダーを選出できません。このタイプのエラーが発生した場合は、上記の条件を確認してください。
- ソケットエラー: 接続数が多すぎる場合に発生します。これを解決するには、kubectl delete <pod_name>を実行して、Podを再起動します。Podは自動的に再作成されます。

Kafkaログでよく報告されるエラーと警告

- ブローカーがIDを登録できない: これは、同じIDを使った複数のブローカーが存在することが原因です。めったに発生しない状況ですが、クラスターでノードの追加や削除を行う際に、クラスターが正しく定義されていない場合に発生する可能性があります。
  この状況が問題であるかどうかを確認する方法: Kafkaブローカーが実行されている各システムに接続し、それぞれに割り当てられたbroker.idの値を確認します。Kafkaノードのbroker.idには、一意の値を定義する必要があります。

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

  version=0
  broker.id=1001

SSL接続エラー

Kafkaとコンシューマーまたはプロデューサー間の接続に問題が発生すると、この警告が報告されます。

- 他のブローカーと通信できない: このようなエラーが発生する場合は、ホスト名が正しく設定されていない可能性があります。ノードがリバースルックアップを実行できないか、DNSが正しく設定されていないことが考えられます。
- 空きディスク不足 (ログまたはデータ): <実際のエラーを示す必要がある>: このような状況が起きないように、保持サイズを適切に設定することが重要です。1つのノードでエラーが発生し、他のノードが問題なく動作している場合は、そのノードのファイルを削除し (クラスター内の他のノードはイベントフローの処理を継続)、該当するトピックに関する保持サイズを手動で変更してから、Podを再起動します。

  また、クラスターにノードを追加することで、個々のノードに保管されるデータ量を削減する方法もあります。この方法を使用する場合は、サポートまたはサービスにお問い合わせください。

  例: あるKafkaノードがダウンし、それに続いてSchema Registryがダウンします。Schema Registryが再起動するには、すべてのKafkaノードが稼働している必要があります。そのため、いずれかのKafkaノードがダウンしている間は、Schema Registryは再起動されません。

- 最初に、Kafkaノードを復帰させる必要があります。Schema Registry Podが起動するには、すべてのKafka Podが実行中である必要があります。Schema Registryは、依存関係が満たされるまで、依存関係にあるPodが実行されているかどうかを確認し続けます。この確認の実行中にSchema Registryがクラッシュした場合、KubernetesによってSchema Registry Podが再起動され、Schema Registryは確認を続行します。

- 結果: Schema Registryがダウンしている間は、変換ストリームプロセッサー (C2AV) は機能しません。Schema Registryが長期間ダウンすると、メッセージが処理 (Avroに変換) されないため、eb-cefトピックのメッセージキューが大きくなります。トピックの保持ポリシーが十分な大きさに設定されている場合、これは問題にはなりません。cefトピックのメッセージキューが増大し続けても、イベントが削除されることはないためです。トピックのサイズまたは時間範囲が保持ポリシーを超えると、トピックから古いメッセージが削除されます。

- Kafkaノードがダウンしたままの場合は、Schema Registryのyamlファイルを編集します (これは高度なタスクであるため、サービスまたはサポートに依頼して実施する必要があります)。

EBの展開を初めて行う際にEBの定義済みトピックが作成されない

- 症状: Bootstrap Web Serviceのログに500応答コード (Schema Registryからの応答) が記録され、トピックが作成されない。

- 回避方法: Event Brokerコンテナーを展開解除してから、再度展開します。

1つまたは複数のコネクターがKafkaにデータを送信できない

- コネクターの接続が正しく設定されているかどうかを確認します。

- コネクターとEvent Brokerの暗号化モード (TLS、TLS+FIPS、TLS+CA、TLS+FIPS+CA) が同じであることを確認します。

- システム上のKafkaポートに接続でき、ネットワークに問題がないことを確認します。

- 接続時に証明書エラー (「証明書を取得できません」) が発生する場合。

    - データパイプラインのすべてのシステムで時刻が同期されていることを確認します。

- Kafka Podがダウンしているかどうかを確認します。1つのブローカーアドレスのみを使用してコネクターの設定を行い、そのブローカーがダウンしているのか、複数のブローカーが存在することを想定しているのかを確認します。後者の場合は、コネクターですべてのブローカーをカンマ区切りリストとして設定する必要があります。

- レプリケーション係数が1に設定されていて、Kafkaブローカーがダウンしている場合、EBを介してデータを送信できません。Kafkaブローカーの問題を修復する必要があります。一般的に、トピックの設定でレプリケーション係数を2以上に指定することで、この問題を回避できます。

- Kafkaの再同期に時間がかかっている: これにより、イベントスループットの低下が起きる可能性はありますが、イベントフローが停止することはありません。

- ディスクが一杯かどうかを確認します。

VerticaでKafkaからのイベントを読み込むことができない

- 新規セットアップ: Vertica Kafkaスケジューラー: Kafkaスケジューラーで、通信ポートとしてKafkaポート39092が設定されていることを確認します。

- 最初は動作していたが、停止した: オフセットが認識されない: このシナリオでは、Kafkaスケジューラーはトピック内のメッセージのオフセットIDを認識できません。これは、Kafkaスケジューラーがトピックからの読み込みを予期しない形で停止し、その後再起動された場合に発生します。
  解決方法: kafka_scheduler deleteコマンドを実行して、メタデータを削除します。コマンドの実行後、すぐにkafka_scheduler createコマンドを実行して、スケジューラーをセットアップします。
- 新規セットアップ: ネットワーク接続を確認します。
- 新規セットアップと既存のセットアップ: ブローカーがダウンしているかどうかを確認します。
- 既存のセットアップ: コンシューマーの接続先のトピックを含むすべてのブローカーを設定していないこと、およびそのコンシューマーで設定されているブローカーがダウンしていることが考えられます。
- 新規セットアップ: SSL接続に関連するエラーが発生した場合、EBとコンシューマーの両方について、証明書をインポートするのに使用した手順を確認します。

EBコンポーネントがクラッシュする: ArcMC Rest API、ルーティング ストリームプロセッサー、変換ストリームプロセッサー

- 起動時: コンテナーの起動順序 (上記) を確認します。依存関係にあるPodに、開始されていないものやクラッシュしたものがあるかどうかを確認します。
- メモリ: JVMでは、システムに実装されているより多くのメモリが必要です。
- すべて: 開いているソケットが多すぎないかどうかを確認します。

EBのEPSが想定値よりも小さい

- ブローカーにリソースの制約があるかどうかを確認します: CPU、メモリ、ディスクが一杯。システムレベルで (またはArcMCで) 使用状況を確認します。
- ネットワークのボトルネック。
- ストリームプロセッサーが変換処理に対応しきれていません。ストリームプロセッサーに何らかの (リソースの) 制約があります。ArcMCで確認すると、ストリームプロセッサーのメトリックがコネクターのEPSよりも小さくなっています。

  十分なリソースがあることを確認します。
- 期待されるファイルシステム
- メモリ
- CPU
- ファイル記述子のロード

FAQ

Event Brokerの展開を構成するKubernetesのPodはどれですか?

- Event Broker Pod
    - confluentinc Pod: kafka、schemaregistry、zookeeper
    - EB Pod: c2av-processor、kafka-manager、orches、routing-processor、web-service

関連トピック:「ArcSight Event Brokerの前提条件」

ArcSight Event Brokerの展開を構成するKubernetesのPodはどれですか?

- Hercules Pod: management、proxy、rethinkdb、search

関連トピック:「ArcSight Event Brokerの前提条件」

KafkaおよびApache ZooKeeperの詳細は、どこで確認できますか?

KafkaおよびApache ZooKeeperの詳細については、以下のリソースを参照してください。

- 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

既存のEvent Broker v1.0を、ArcSight InvestigateおよびVerticaと組み合わせて使用することはできますか?

いいえ。ArcSight Investigateには、Event Broker 2.0が必要です。Event Broker 1.0からのデータの移行には、Event Brokerデータ移行ユーティリティを使用できます。このツールの入手については、ArcSightサポートにお問い合わせください。

関連トピック:『Investigate 1.0 Deployment Guide』。入手可能な場合は、『Event Broker Data Migration Tech Note』も参照してください。

Event Brokerのインストール方法は1つだけですが、これらの各PodをKubernetesのワーカーノードに配布するにはどうすればいいですか?

ArcSightインストーラーアプリケーションによって、Kubernetesワーカーノードの配布と展開が自動的に処理されます。

関連トピック:「Kubernetesノードの追加」

EBがインストールされるマシンのホスト名の確認方法を教えてください。

例:

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

NAME              STATUS    AGE       FQDN

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

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

上記のEBでラベルを確認すると、マスターノードとワーカーノードで同じラベルが使用されていることがわかります。これらが正しいラベルであることを確認する必要があります。

ホスト名が正しく設定されているかどうかを確認するには

```
# 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.
```