ホワイトペーパー

あらゆるソースからのデータを どこにでも活用: オープンアーキテクチャー

MICRO FOCUS

目次

エグゼクティブサマリー 3

最新のセキュリティ監視 3

何が役立ち何が役立たないか 5

オープンスタンダードとオープンアーキテクチャー 5

オープンアーキテクチャーの活用とユースケース 7

オープンアーキテクチャー内のユースケース 7

Elastic、Splunk、およびHadoopの利用 9

付録A: Elastic 10

付録B: Splunk 13

付録C: Hadoop 15

まとめ 25

企業情報 26

エグゼクティブサマリー

大量のデータセットは、企業がデータの収集および分析を行う方法に変化をもたらしています。このデータを分析し、有効なビジネス上の判断を行うための専用の分析ツールやアプリケーションでは、オープンアーキテクチャーを使用する必要があります。Open ReferralのWebサイトには、テクノロジーの「オープン性」に関する説明として、「オープンプラットフォームにより、ユーザーがプラットフォームのデータに直接アクセスできるだけでなく、オープンな形式でデータをパブリッシュすることも可能になります」と記されています。[1] オープンテクノロジーを導入することで、ニーズに合わせてアプリケーションやインフラストラクチャーとの間でデータを容易に移動できるようになります。共通データモデル、形式、および標準プロトコルをサポートしたテクノロジーの実装は、企業環境におけるオープンアーキテクチャーの代表例です。オープンプラットフォームでは、各種ツールセットの柔軟な活用が可能で、それぞれの用途に最適なツールを利用できます。大量のデータを通じて、高い生産性と卓越したビジネス価値が実現されます。閉じたシステムではデータの共有が難しく、データのサイロ化が起きます。一部には、オープンスタンダードやオープンテクノロジーと互換性を持たせないように製品を設計しているベンダーも存在します。こうした「ベンダーロックイン」戦略では、カスタマイズしたツールの活用が進み、競合製品への切り替えが高コストで困難なものになります。このベンダーロックイン戦略によって、これまで企業は現状を維持せざるを得ませんでした。ときには、これが企業の成長、業績、安定性を損なうこともありました。

多くのデータプラットフォームでは、さまざまなユースケースに対応するために複数のデータベースが必要になります。場合によっては、複数のビッグデータプラットフォームが必要になることもあります。1種類のプラットフォームをあらゆるユースケースに利用できるという考え方は、誤解を招く間違った考え方です。データやテクノロジーがインテリジェントな意思決定に役立つように進歩していくのに合わせて、さまざまな形式、システム、およびテクノロジーを使用する必要があります。さまざまなデータリポジトリ、システム、およびツールを使用する最新のセキュリティオペレーションセンターの場合は特にそうです。

オープンアーキテクチャーの仕様は、正式[2]に承認された規格を通じて公開されています。また、プライベートで設計されたアーキテクチャーで、仕様が公開されたものもあります。このように仕様が公開されているため、他のテクノロジーと柔軟に統合することができ、フロントエンドのアプリケーションと分析用インフラストラクチャーとの間でデータを容易に移動することができます。企業はオープンアーキテクチャーを導入する時点では、コストを抑えて投資効果を得ることを見込んでいますが、新しいシステムとの統合や別のインフラストラクチャーへの移行を行う段階になって、オープンアーキテクチャーに別の価値があることに気付きます。

最新のセキュリティ監視

セキュリティオペレーションセンター (SOC) を設置したり、マネージドセキュリティーサービスプロバイダー (MSSP) を利用したりしてセキュリティイベントやアラートを監視している企業も、同様の課題や問題に直面しています。1つ目の問題点は、実装可能な標準ソリューションとして即稼働できる完璧なセキュリティオペレーションチームやマネージドサービスが存在しないということです。誰もが予算の制約と限られたリソースの中で業務を行っています。また、多くの場合、組織構造やレガシーインフラストラクチャーによって状況はさらに複雑なものになっています。MSSPは、一定の制限の中で、専任のスタッフが24時間体制で最高レベルのセキュリティ監視を提供しています。外部組織のセキュリティイベントを監視するという業務の性格から、一部の企業では、MSSPが実施できる業務をアラートのエスカレーションと推奨事項の提案に限定し、インシデントの修正を行えないようにしています。社内に設置されたSOCでは、さらに詳細な機能、プロセス、およびツールを用いてインシデントに対応することができます。すべての企業に、トリアージ、マルウェアの分析、検出、およびインシデント対応を行うための要員とツールが揃っているとは限りません。SOCの運用を成功させるには、複雑なセキュリティユースケースを解決するための適切なツールを揃える必要があり、さまざまなツール、テクノロジー、およびメカニズムを柔軟に組み合わせて使用する必要があります。セキュリティ違反の事後調査が必要なインシデントが発生した場合には、データを取り出す機能が不可欠です。

[1] openreferral.org/faqs/what-do-we-mean-by-open-platforms-whats-an-api/

[2] openreferral.org/documentation/

2つ目の問題点は、テクノロジーを利用することでスタッフやプロセスの増強が必要になることです。セキュリティ製品を購入してもアナリストのトレーニングや製品の構成とチューニングに時間を掛けていない場合、セキュリティが煩わしい存在になり、大量の誤検知が発生することになります。適切に設定されていないデバイスからは、有効なデータがまったく送信されない場合もあります。適切にインストール、構成、およびチューニングされていない場合、システムから大量のアラートが生成されて、アナリストがデータを解析できなくなります。Rawデータは、状況によっては有効ですが、セキュリティユースケースではあまり役立ちません。コンテキストや状況を把握することが非常に重要になります。また、データのエンリッチ化も重要です。欠落データを補足し、データが正しいことを確認し、クリティカルな情報を追加することが重要です。たとえば、Microsoft Windowsのログでは、細かな情報が冗長に出力される傾向がありますが、その一方で、一部のログではIPアドレスやホスト名などの重要情報が欠落することがあります。この場合、状況はさらに複雑になり、ログの意味を理解するための追加処理が必要になります。コンテキストを理解してログデータをエンリッチ化する高度なログ収集プロセスを用意することで、大量のデータセットを理解して処理するのが容易になります。テクノロジーを利用することで、セキュリティチームがそれまでよりも容易にかつ効果的に業務を行えるようにする必要があります。高額な市販のソリューションを購入する前に、自動化によって連携させた複数のオープンソースソリューションを使用して概念検証を実施し、検討対象のユースケースの価値を明確にする必要があります。

3つ目の問題点は、アラート疲れです。これは、エラーや人材不足の原因になります。アナリストにアラートのトリアージを専任で担当させた場合、仕事内容は重要でも、非常に退屈でつまらない業務になる可能性があります。多くの優秀なアナリストは、セキュリティ投資の計画と優先順位付けが経営幹部によって適切に行われている職場に移ってしまうでしょう。また、アラート疲れは人的ミスにもつながります。1日に何十回または何百回と行われるトリアージプロセスでは、手順に漏れが起きても不思議ではありません。最高の人材を雇って1日中アラートのトリアージを行わせるセキュリティチームは、持続可能な管理を実現するための環境の整備を怠っています。可能であれば、自動化によってトリアージに伴う手動業務を軽減するか、アラートのトリアージを別の部門に振り分けて、調査業務を行う上級セキュリティアナリストの負担を減らす必要があります。優れたチームは、セキュリティアナリストに正確さと効率性を高めるためのアイデアを提供させ、実際にワークフロー、プロセス、およびツールの変更を行う権限を与えています。

最後の問題点は、コンテキストのないアラートが実用的でないことです。多くの場合、セキュリティ監視は構成されたアラートに基づいているため、必要なデータの断片のみが、コンテキストに関するあいまいなシグネチャとともに取得されます。これらのアラートの一部が正しい検知であっても、確認できる証拠がなければ、これらのアラートは誤検知に分類される可能性があります。フルパケットキャプチャー、ホストフォレンジック、エンドポイントログ、アプリケーションサーバーログ、ネットワーク分析データ、エンティティ行動、およびその他のコンテキストを含むネットワークセキュリティ監視手法を使用することで、SOCは企業のセキュリティをより効果的に監視することができます。

このデータをすべてSIEMに取り込むことは困難で、スケーラブルに行うのは不可能な場合もありますが、企業のセキュリティ保護における重要な目標の1つであることに変わりはありません。複数の異なるシステムのデータとログを保管することで、システムの動作、トランザクション、およびユーザーとシステムとの間のインタラクションを把握するのに使用できる監査証跡を残すことができます。このような異なるシステムのデータを組み合わせることで、統計分析を行い、さまざまなシステムの一見無関係なイベントの間に相関関係を見いだすことができます。セキュリティデータを分析して得た新しい発見が、セキュリティ上の価値だけでなくビジネス価値にもつながります。

オープンアーキテクチャーは、インフラストラクチャーを最大限に活かし、さまざまなアプリケーション間でのデータ活用を可能にします。分析用のデータレイクとの接続や、革新的なビジネス分析を行うための他のアプリケーションとの接続は、オープンアーキテクチャーに合うように設計されています。データを任意の場所から任意の場所へ安全に移動する機能は、最初に行うべき設計判断です。その後は、企業内でセキュリティ状況やイベントに関するデータを柔軟に活用し、特定の環境やニーズに合ったソリューションと統合することができます。

何が役立ち何が役立たないか

データは、量が多ければ多いほど良いとは限りません。セキュリティ違反が発生した後に手掛かりをつかむため、部分的に整理された形で利用可能なすべてのデータを収集するようにシステムを設計することは、セキュリティアーキテクチャーの構築時に起こしやすい間違いです。この方法には2つの問題があり、無関係なデータによるアラート疲れに逆戻りしてしまう上、収集プロセスにコンテキストが欠落しているため、収集したデータがセキュリティ違反を再現するのに役立ちません。成果を上げている企業では、収集する内容を判断する際にデータストリームの価値を考慮しています。長期的なストレージコストという観点で、データストリームを精緻化することが投資収益の増大につながります。非セキュリティデータにセキュリティコンテキストを追加してエンリッチ化することで、これまでよりも迅速かつ正確に脅威の検出を行うことが可能になり、セキュリティの専門知識をRawデータに適用することができるようになります。また、オープンアーキテクチャーにより、さまざまなセキュリティユースケースで、任意の場所からのデータ収集が可能になります。

データ量が多ければセキュリティが向上するとは限りません。ログの意味や状況を理解していなければ、ハッカーを見つけることはできません。処理しきれない大量のアラートが発生したり、コンテキストが欠落したりしている状況では、データが増えるほど、それに対応するのに多くの要員が必要になります。通常の対応策から外れたアラートは、調査を行うことになります。これらは疑わしいアラートである可能性があるため、調査する必要があります。たとえば、Windows ServerでパッチをインストールするのにAdministratorというユーザー名が使われている場合、正規のログまたはアラートかもしれませんし、ハッカーによる攻撃かもしれません。いずれにしても、調査が必要になります。そのため、データ量が増えるほど、こうした状況も増えることになります。テクノロジーを使用して分析を行う場合でも、この疑問に答えるためのインテリジェンスを追加するのに人員が必要になります。このため、人員数も調査量も増えますが、たいていは十分な対応を行うことができずセキュリティレベルが低下します。

オープンアーキテクチャーでは構造化データによってコンテキストが提供されるため、アクションの属性を重要なフィールドに格納し、これを統合されたツールセットで利用することができます。構造化されたセキュリティデータを保管することで、過去の動作や属性に基づいた複合的なアラートの利用が可能になります。データを構造化しておくことで、企業はデータマイニングやデータ分析にデータを活用できるようになります。[3] 構造化されたデータは、検索エンジンアルゴリズムやその他の検索操作ですぐに検索できるようにすることができます。構造化されていないデータは、まとめるのに長い時間と労力が必要になります。

オープンスタンダードとオープンアーキテクチャー

さまざまな形で実装されるエンタープライズアーキテクチャーの重要な要素によって、非常に大きなデータセットの利用という共通のビジネスニーズが解決されます。ApacheのKafka[4]、Spark[5]、およびHadoop[6]などのテクノロジーでは、オープンスタンダードを利用することで、データの処理や他のプラットフォームとの情報のやり取りを柔軟に行うことができます。オープンな標準ベースのアプローチにより、既存のインフラストラクチャーにすばやく導入できるということもあり、これらのテクノロジーは幅広く採用されています。このテクノロジースタックは、現在の企業によってほぼ際限なく生み出されるデータセットを処理する安定したフレームワークを提供します。オープンスタンダードを利用することで、セキュリティプロバイダーは、企業のセキュリティを維持するという際限のない課題の解決に注力することができます。

セキュリティデータは、当然ながら企業にとっての機密データとして分類されます。機密データをまとめることで、関連するデータストア自体のリスクレベルが高くなり、攻撃や事前調査のターゲットになります。このため、これまではセキュリティデータは意図的に、必要に応じて小分けにしてサイロに保管されていました。必要に応じてセキュリティログを生成する場合には、時間がかかる可能性があります。

[3] psnow.ext.hpe.com/doc?id=4aa6-5850enw.pdf
[4] kafka.apache.org/
[5] spark.apache.org/
[6] hadoop.apache.org/

アウトソースされている場合は特にそうです。今日の変化の速いビジネス環境では、セキュリティデータがすぐに利用できない場合、十分なセキュリティデータなしに意思決定を行うことになります。

データは収集する時点で、構造化された価値のあるものでなければなりません。ログ、センサー、ストリームネットワークトラフィック、セキュリティデバイス、Webサーバー、カスタムアプリケーション、クラウドサービス、およびその他のセキュリティデータは一旦記録されると、その後は変更できなくなります。変更不可能なデータは、検証可能な形で変更されずに維持されます。これは、データベースからセキュリティ違反を再現するのに不可欠な属性です。また、大量のセキュリティデータセットから重要なビジネス情報をマイニングすることもできます。たとえば、従来型IT、エンドポイント分析などのセキュリティツール、SalesForceなどのクラウドIT、バッジ読み取り装置などのIoT/OT、またはiOSやAndroidなどのモバイルで、Active Directoryから人の動きに関する情報などを取り出すことができます。

ArcSight Data Platform (ADP) アーキテクチャーは、これらのオープンスタンダードを活用して企業全体でセキュリティデータを利用できるようにすることで、チーム、プロジェクト、およびビジネスユニットの垣根を超えたビジネスニーズやセキュリティニーズへの対応を可能にします。

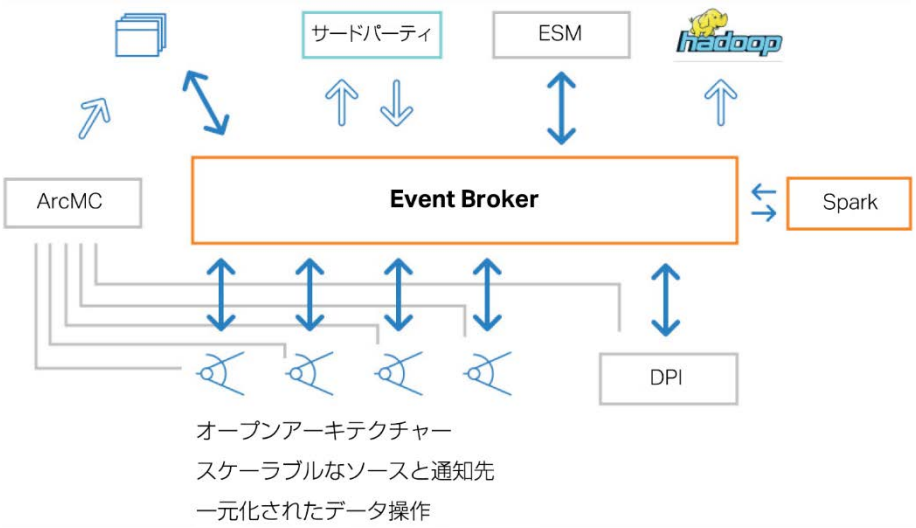


図1: ArcSight Data Platform (ADP) の機能

ArcSight Data Platform (ADP) は、Hadoopを含むサードパーティ製プラットフォームとシームレスに連携する業界初のオープンセキュリティアーキテクチャーを実現します。ADPコンポーネントには、以下が含まれます。

Event Broker: KafkaベースのEvent Broker (EB) では、毎秒最大100万イベントの処理が可能です。EBはメッセージバスを提供するADPのコンポーネントで、複数の通知先の間でのデータのスケーラブルな分配に対応しています。EBを利用することで、企業内の複数のシステムやテクノロジーの間で、ログデータなどのデータの受信と分配を容易に行うことができます。また、EBはスケーラビリティや耐障害性にも優れています。

ArcSight Management Center: ArcSight Management Center (ArcMC) では、エンドツーエンドの監視を一元的に表示し、一括操作の簡素化された処理を行うことができます。

Logger: Loggerは、高いイベントスループット、効率的な長期記録、高速なデータ分析に最適化されたログ管理ソリューションです。

SmartConnector: あらかじめ構築された350を超えるコネクターを利用して、手動でカスタマイズすることなく、データ収集ソースを容易に拡張することができます。データにコンテキストや意味を付加してエンリッチ化することで、データを関連性のあるものにします。

ロードバランサー: SmartConnectorロードバランサーは、SmartConnectorのステータスと負荷を監視することで、「コネクタースマート」なロードバランシング機能を提供します。

ADPはデータ収集プロセスに変革をもたらし、さまざまなデータソースから高速かつ大量のデータ取り込みを可能にすることで、管理タスクを簡素化し、監視をより効果的なものにします。Kafkaは、毎秒数百メガバイトで数千のクライアントを処理できるように設計されたApacheのオープンソースプロジェクトです。Kafkaでは、企業が必要としているクラスター化された高可用性 (HA) 環境内で大規模にデータが分配されるため、EBの基盤テクノロジーとして採用されました。

ADP EBはイベント処理を一元化し、ArcSight環境のスケーリングを容易にし、ArcSightのセキュリティデータを他のアプリケーションやサードパーティ製ソリューションで利用できるようにします。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、優れた拡張性とスループットを備えたマルチブローカークラスターの利用が可能になります。

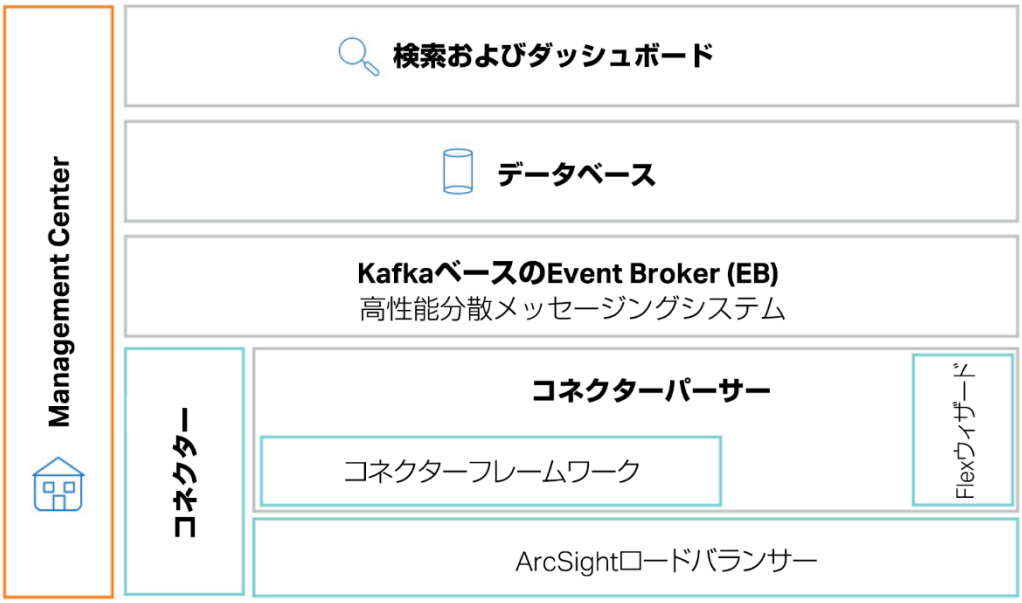


図2: ArcSight Data Platform (ADP)

ADP EBには、Apache Kafkaのプリパッケージバージョンが含まれています。EB (またはブローカーのクラスター) をインストールして設定すると、ADP SmartConnectorはセキュリティデータのパブリッシュを開始します。ADP LoggerおよびApache Hadoopまたはその他のコンシューマーは、このセキュリティデータをサブスクライブできます。

オープンアーキテクチャーの活用とユースケース

オープンアーキテクチャーを活用することで、ADPはデータ消費とデータ分配に変革をもたらす最前線になります。独自の拡張性と、企業向けの強力なログおよびイベント分配システムにより、ADPは将来の戦略的セキュリティプラットフォームとなります。ArcSight、Hadoop、および企業の社内セキュリティデータウェアハウスとの間で送受信されるデータに対し、フィルターの適用、コントロールの追加、またはインストリーム処理を行うことができます。Micro Focus® ADPは、さまざまなユースケースに対応した柔軟な形式と変換を通じて、エンタープライズクラスのセキュアデータ環境を実現します。

オープンアーキテクチャー内のユースケース

HIPAA準拠と患者のプライバシー

救急クリニック、ラボ、診療所、およびヘルスケアネットワークは、組織内に多様なシステムやさまざまな機器が接続されているため、情報の保護が特に困難です。これらのネットワークは、ネットワーク上を流れる機密データのセキュリティを維持しながら、競争力を維持するためにネットワークに最新のテクノロジーを追加するという困難な状況に直面しています。これらのネットワークでは、PCI DSSとHIPAAに基づいてセキュリティを確保する必要があり、カード所有者のデータや患者の医療情報を保護するための幅広いリスク軽減手段が必要になりま

す。ADPはサードパーティ製アプリケーション上のデータを活用し、コンプライアンスパックを介してHIPAAに関する問題を把握することで、幅広い領域をカバーし、プライバシー侵害の検出時にすばやく対応することができます。

悪意のある内部者と脆弱なエンドポイント

石油/ガス探査業者やガス管メーカーは、多くの場合厳しい環境で、重要な遠隔測定データを処理しています。ArcSightの幅広い製品群は、分散環境でこのデータを保護するのに役立ちます。データが遠隔地にあるかデータセンター内にあるかに関係なく、さまざまなモデルで個々のデータのニーズに対応することができます。ADPのオープンアーキテクチャーで、成熟したインフラストラクチャーによるセキュリティオペレーションセンターを構築することで、あらゆるソースからのデータを幅広く利用することが可能になります。ADPでこのデータをログに記録し、ArcSight ESMでセキュリティ分析を行って攻撃者を識別します。

クレジットカード業界データセキュリティ基準 (PCI DSS)

小売業界の中小企業は、米国の場合、市や州によって異なるネットワーク上のデバイスを管理しなければならない複雑な状況にあります。PCI DSSに準拠していない場合の規則やペナルティを考慮すると、PCI DSS準拠を簡単に監査および管理できるようにしておくことが不可欠です。ADPのSyslogイベント管理とコンプライアンス問題に関するレポート機能を利用すると、クレジットカード情報およびPOSシステムを保護できます。ADPでArcSight Compliance Insight Package for IT Governanceを使用すると、組織は、継続的なイベントおよびログデータのレビュー、ログのオンラインおよび長期アーカイブでの保管、フォレンジック分析への対応、監査への対応、および準拠状況の証明を自動的に行うことができます。ArcSight Data Platformは、ログ管理を経済的に自動化できるオールインワンのログ収集・保管・分析アーキテクチャーで、既存のデータリポジトリやデータレイクとも接続できます。

米証券取引委員会 (SEC) および米連邦取引委員会 (FTC)

信用組合、ペイデイローン (PayDay Loan) サービス会社、および中小金融機関はいずれも、SECおよびFTCの法規制準拠に対応する必要があります。Micro FocusのArcSightおよびデータセキュリティ製品には、脅威の管理に加えて、ユーザーおよびアプリケーションアクセスを管理するためのプラットフォームが用意されています。環境のロックダウン機能とADPのログ機能を使用することで、大量の履歴データを利用して、分析、コンプライアンス、および内部監査の支援を行うことができます。

注目のユースケース: 悪意のある内部者と脆弱なエンドポイント — 禁止されたWebサイトでのプロキシログ分析

リソースアクセスを追跡して内部者による悪用や不正行為を見つけることは、一般的なセキュリティユースケースです。この情報は、インシデント対応時に攻撃者がアクセスして改変を行った可能性のあるリソースを特定するのに役立ちます。調査方法の1つに、禁止されたサイトや既知のマルウェアソースにアクセスしようとしてプロキシでブロックされた回数の多い内部ユーザーを調べる方法があります。このWebアクセスデータレポートは、セキュリティ被害を受けたシステムの追跡、データ漏洩の追跡、排他的な証拠に基づく生産性の向上など、さまざまな用途に使用できます。

ADPはオープンアーキテクチャーであるため、他のプラットフォームやアプリケーションとの間でデータを移動して分析を行うことができます。このため、調査内容に合わせて追加のツールを使用することが可能です。多くのSOCで、ELK、Splunk、Hadoopなどのさまざまなツールが活用されています。リソース支出の妥当性の証明、セキュリティ違反発生後のレビュー用の最終的な監査証跡の提供、インシデント対応計画の見直し、ハントポートフォリオの拡充など、どのような目的でセキュリティデータにアクセスする場合でも、オープンスタンダードを通じて適切なツールを使ってデータにアクセスすることができます。

以下では、他のプラットフォームと組み合わせてADPとEBを使用する手順を示します。

Elastic、Splunk、およびHadoopの利用

Elastic

Elasticプラットフォームは、ログの管理と分析に使用されます。このユースケースでは、単一ノードのElastic 5.0サーバーをセットアップし、Logstashを介してEBからADPイベントを取り込み、Kibanaを使用してデータの可視化 (ビジュアリゼーション) を作成します。

1. ElasticのWebサイトに記載されている手順を使用して、Elasticをインストールして構成します。[7]
2. イベントに構造を適用する場合は、リッチCEFマッピングに対応した、共通イベントフォーマット (CEF) コーデックプラグインをダウンロード[8]して使用できます。また、独自のCEFフィルターを手作業またはプロフェッショナルサービス契約を通じて作成することもできます。(CEFフィルターの例は、付録Aに記載されています。)
3. Kibanaをインストールして構成します (付録Aを参照)。
4. 初期インデックスを作成し、すべてのフィールド (CEFフィールドかどうかに関係なく) が検索とアグリゲーションに使用できることを確認します。これは、検索とビジュアリゼーションを行うのに必要です。
5. 次に、Logstashでイベント取り込みからCEFフィールドを抽出します。
6. プロキシデバイスから拒否された回数の多い上位ユーザーなどのビジュアリゼーションを表示できます。

Splunk

Splunkは、デバイスから収集した機械生成データのキャプチャー、インデックス作成、検索、および分析を行うデータプラットフォームです。ADPのオープンアーキテクチャーでは、イベントがSplunk環境に流入することが可能です。

1. SplunkのWebサイトに記載されている手順を使用して、Splunkをインストールして構成します。[9]
2. Kafkaメッセージング用の新しいデータ入力を構成します。
3. KafkaポートではなくZooKeeperポートを指定します。ZooKeeperホストで、EBの最初のノードを指します (付録Bを参照)。
4. イベントがSplunk環境に流入するのがわかります。

Hadoop

Hadoopは、分散コンピューティング環境で大量のデータセットの処理に対応できるように設計されたJavaベースのプログラミングフレームワークです。データの増加は多くの企業戦略の重要な要素になっていますが、Hadoopは優れた効率性によって最新化に伴うコストを軽減してきました。ADPはADP Event Brokerを介してHadoopと連携することで、セキュリティデータをビジネスアプリケーションで利用できるようにします。

Cloudera HadoopとADPとの連携には、複数のサービスが使用されます。たとえば、イベントストレージ用のHDFS、ビジュアリゼーションおよびクエリ用のHUE、およびADP EBからのイベントをHDFSに取り込むためのFLUMEなどがあります。

1. ClouderaのWebサイトに記載されている手順を使用して、Cloudera Hadoopをインストールして構成します。[10]
2. FLUMEサービスの追加、開始、および構成を行います (付録Cを参照)。
3. HDFSからのイベントの表示するようにHUEを構成します (付録Cを参照)。

---

7 elastic.co/guide/index.html

8 elastic.co/guide/en/logstash/current/plugins-codecs-cef.html

9 splunk.com/en_us/download.html Splunkのインストール手順

10 cloudera.com/products/apache-hadoop.html

4. FLUMEサービスとHDFSサービスが開始されたら、“flume” というディレクトリが表示され、CEFファイルとその内容が格納されます。

付録A: Elastic

Elasticは、一連のアプリケーションとツールで構成された、ログの管理と分析を行うための優れたプラットフォームです。ElasticはSplunkの強力なライバルですが、オープンソースであるため、コミュニティからの強い支持を受けています。Elasticは、Loggerの強力なライバルでもあります。Loggerには、導入して効果を発揮するまでの手順が比較的容易であるという明確な利点があります。反対に、Elasticには、十分な時間をかけて構成できる場合には、非常に柔軟な利用が可能であるという利点があります。

以下の手順は、EventBrokerのトピックを読み込むため、Kafkaプラグインを使用して構成されたLogstashを介してイベントの取り込みを行う単一ノードのElastic (5.0) の場合です。

基本的なElasticノードのインストールと構成は、非常にシンプルです。次のWebサイトの手順に従って、インストールと構成を行います。
elastic.co/guide/index.html

重要なのは、EventBrokerからイベントを取得し、これらのイベントを構造化されたKibanaで簡単に検索できるものに転送することです。以下は、分析を行うために、BlueCoatイベントの基本/必須のCEFフィールドをすべて取得するCEFフィルターです。ただし、リッチCEFマッピングに対応済みのCEFコーデックプラグイン (手動インストール) がelastic.co/guide/en/logstash/current/plugins-codecs-cef.htmlで提供されています。CEFイベントの取得方法として、いくつかの選択肢があることも覚えておいてください。

```
Input {
    kafka {
        topics => [ “BlueCoatEvents","ProductionEvents"]
        bootstrap_servers => “eventbroker:9092"
    }
}
The above code simply pulls two topics from the EventBroker.
filter {
    # Manipulate the message
    mutate {
        # Saved the original message into a temporary field
        add_field => { “tmp_message" => “%{message}" }
        # splits message on the “|" and has index numbers
        split => [ “message", “|"]
        # generate fields for the CEF header
        add_field => { “cef_version" => “%{message[0]}" }
        add_field => { “cef_device_vendor" => “%{message[1]}" }
        add_field => { “cef_device_product" => “%{message[2]}" }
        add_field => { “cef_device_version" => “%{message[3]}" }
        add_field => { “cef_sig_id" => “%{message[4]}" }
        add_field => { “cef_sig_name" => “%{message[5]}" }
        add_field => { “cef_sig_severity" => “%{message[6]}" }
}
```

上記のコードでは、“mutate” プラグインを使用してCEFヘッダーを有用な情報に分解しています。ここからは、CEFがキー/値になります。“kv” プラグインでこの操作を行います。

```
    # Parse the message with field=value formats
    kv {
        # Note: values with spaces are lost (still getting there)
        field_split =>  “ "
        trimkey =>  “<>¥[¥], "
        trim =>  “<>¥[¥],"
        # Only included the fields which are of interest (dont need everything)
        include_keys =>
        [ “cat", “act", “proto", “dst", “dpt", “src", “spt", “dhost", “suser", “request", “catdt"]
    }
```

“kv” はここでは、特定のフィールド名の抜き出しのみを行っています。フィールド名の配列は簡単に拡張して、必要に応じて文字通りすべてのCEFフィールドを取り込むこともできます。

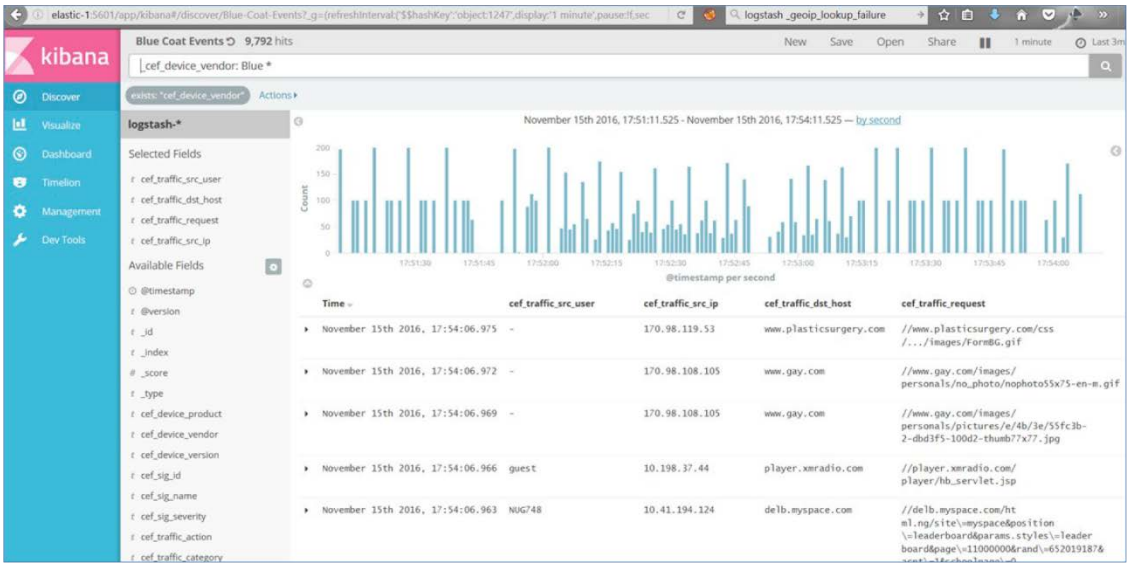
mutateを再度呼び出して、CEFフィールドの名前を変更し、Kibanaで扱いやすくします。

```
    mutate {
        # Rename fields to cef_field_names
        rename => [  “cat",  “cef_traffic_category"]
        rename => [  “act",  “cef_traffic_action"]
        rename => [  “proto",  “cef_traffic_proto"]
        rename => [  “dst",  “cef_traffic_dst_ip"]
        rename => [  “dpt",  “cef_traffic_dst_port"]
        rename => [  “src",  “cef_traffic_src_ip"]
        rename => [  “spt",  “cef_traffic_src_port"]
        rename => [  “dhost",  “cef_traffic_dst_host"]
        rename => [  “suser",  “cef_traffic_src_user"]
        rename => [  “request",  “cef_traffic_request"]
        rename => [  “catdt",  “cef_traffic_category"]
        # Revert original message and remove temporary field replace => {  “message" =>
         “%{tmp_message}" } remove_field => [  “tmp_message" ]
        }
    }
```

最後は、当然ですがElasticにイベントをプッシュする必要があります。ただし、デフォルトで、Elastic (および関連する他のツールのほとんど) は、ループバックのみのバインドと通信を想定しています。そのため、最初にこれを修正する必要があります。

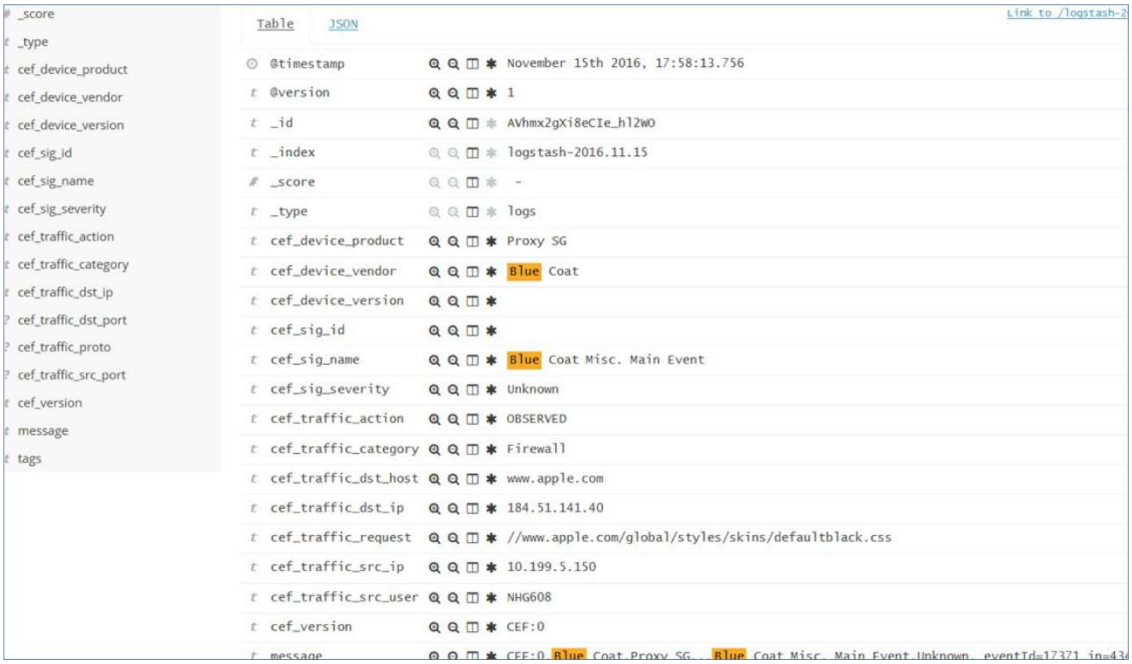
```
output {
    elasticsearch {
        hosts => [ “elastic-1:9200"]
        }
}
```

Kibanaのインストールと構成 (elastic.co/guide/en/kibana/current/install.html) が済んだら、初期インデックスを作成し、すべてのフィールド (CEFフィールドかどうかに関係なく) が検索とアグリゲーションに使用できることを確認します。これは、検索を行い、ビジュアリゼーションを作成するのに必要です。ビジュアリゼーションを行う方法は、次のURLにわかりやすく記載されています。
digitalocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations

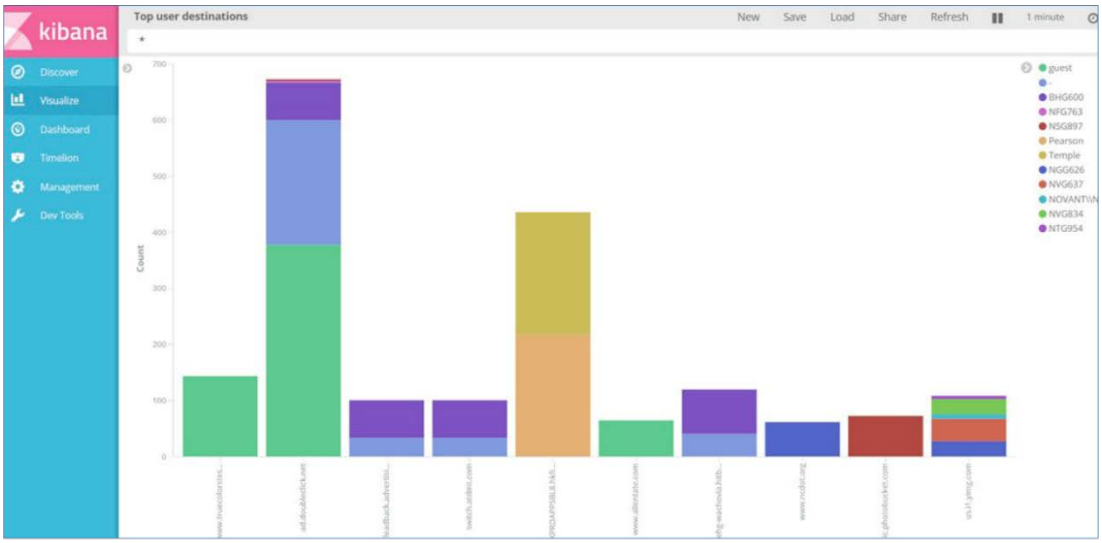


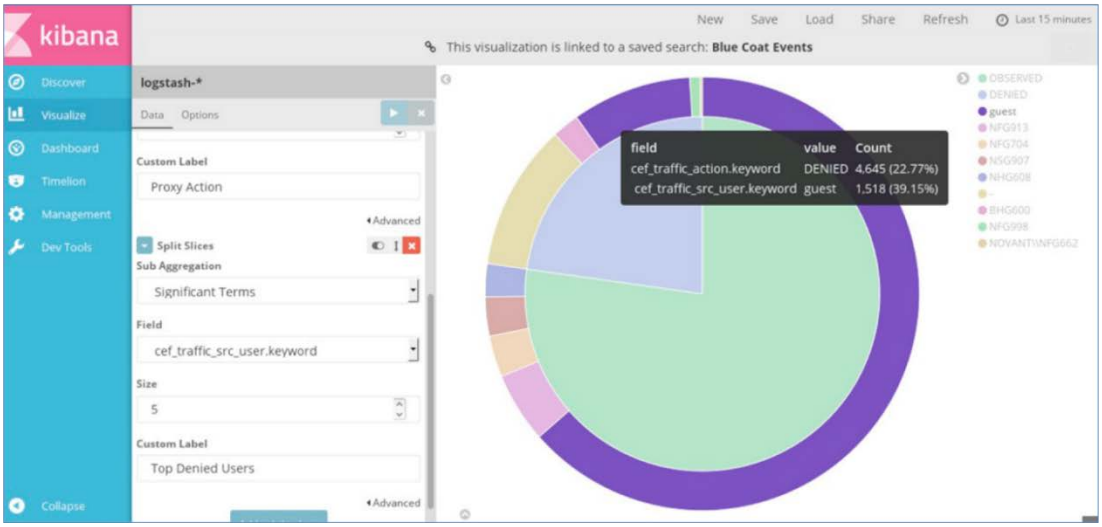
Logstashでイベント取り込みから取得したすべての "CEF" フィールドを参照できます。ここでは、10個程度のフィールドを操作しているだけです。言うまでもなく、CEFには数百のフィールドがあります。上記のURLは、デモ用イベントからのものです。

各イベントを取得して開くと、分析およびビジュアル化が必要な詳細を確認できます。



この後、検索とビジュアリゼーションのデモを行うことができます。以下は、上位の接続先ホストと、これらのホストへの接続要求を行っている関連ユーザーを示す積み上げ棒グラフです (Blue Coat)。





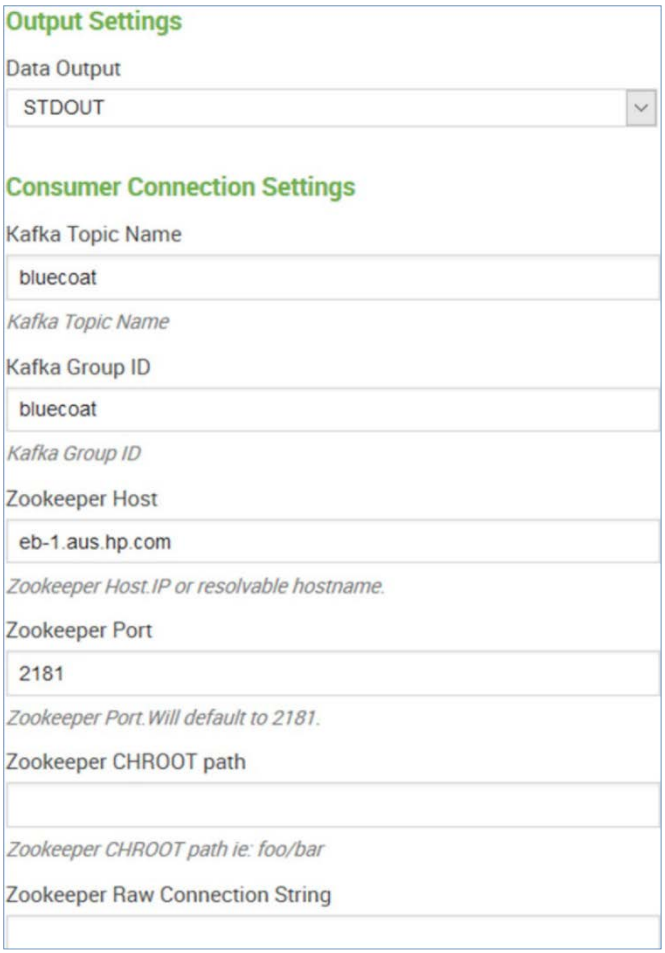
上記のビジュアリゼーションには、監視対象プロキシデバイスから拒否された回数の多い上位ユーザー名が表示されています。

付録B: Splunk

Splunkには、すぐに使用を始めることができるように小規模で利用するためのライセンスが付属しています。規模を大きくする必要がある場合は、開発者ポータル (dev.splunk.com) を参照してください。6か月有効な1日あたり10 GBのライセンスを入手できます。

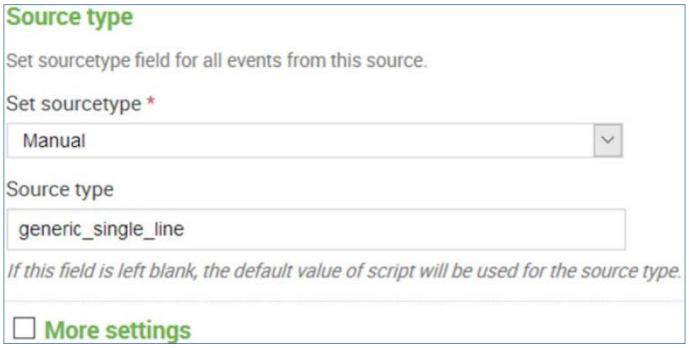
WindowsまたはLinuxにSplunkをインストールします。Windowsの場合、Splunkは1つの実行可能ファイルです。次に、Event Broker (KAFKA) からイベントを取り込みます。これはとても簡単です。

1. [Settings] > [Data Inputs] にアクセスします。
2. [KAFKA Messaging] を選択します。
3. [New] を選択します。
4. 関連するフィールドに入力します。以下のスクリーンショットを参考にしてください。

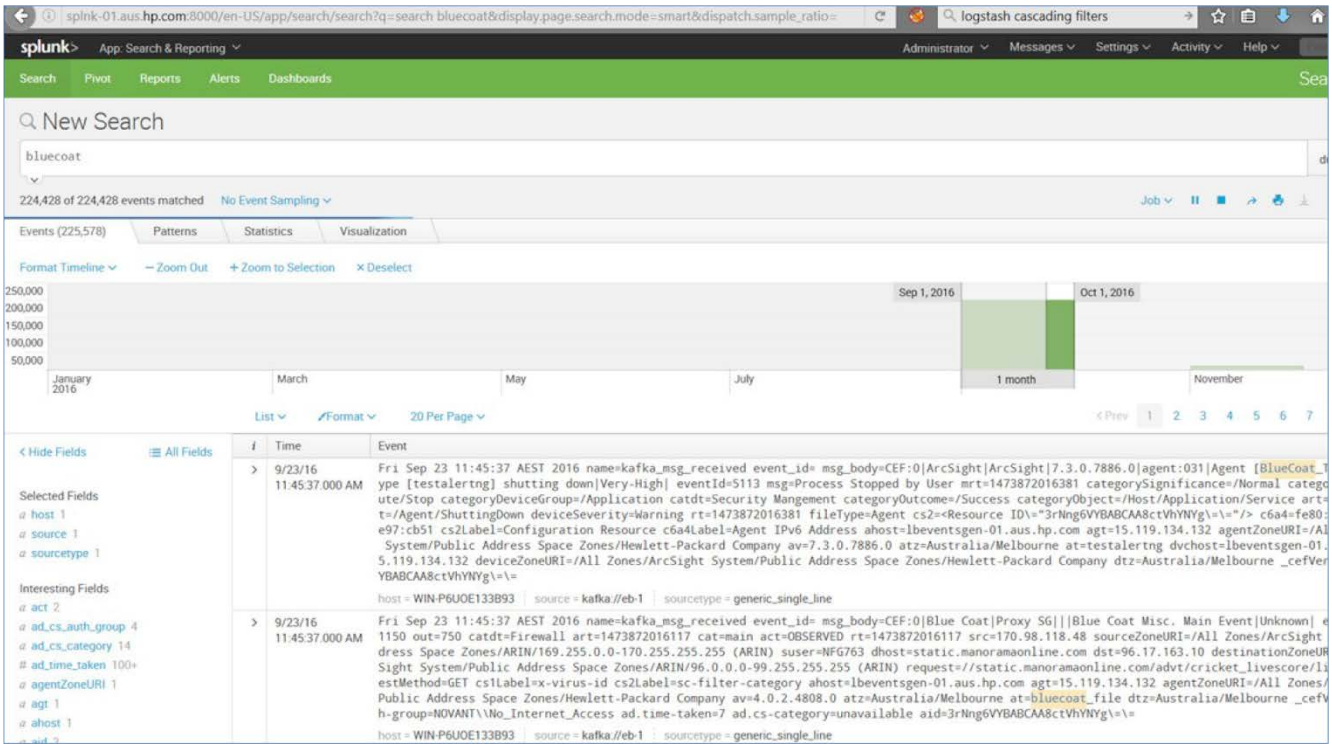


トピック名は該当するトピックです。グループIPは、それぞれの「コンシューマーグループ」です。ZooKeeperホストは、Event Brokerノード#1です。Splunkアプリケーションでは若干古いテクノロジーの実装が使用されるため、必ずKafkaノードポート (9092) ではなく、ZooKeeperポートを指定してください。

ソースタイプには、下のスクリーンショットのように "generic" を使用しました。



この後、Event Brokerからイベントが届き始めるのを確認できます。

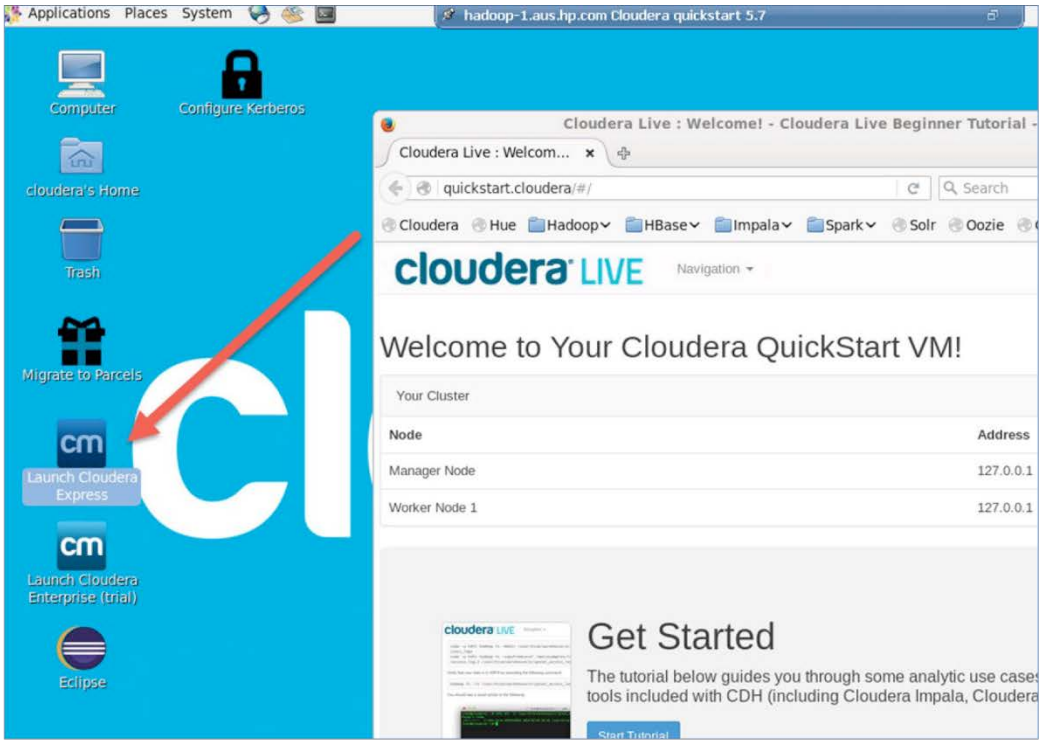


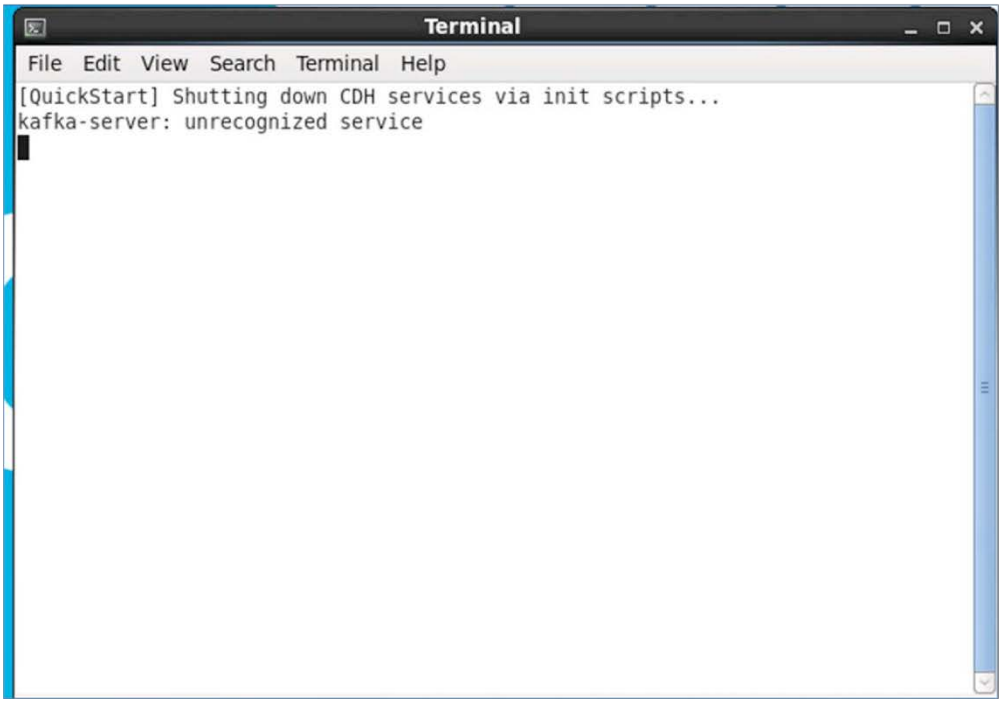
付録C: Hadoop

Hadoopには統合のデモを行うための非常に簡単な方法があり、『ADP Event Broker管理者ガイド』に詳しい手順が記載されています。より洗練され外観的に優れたものにするため、ここでは無償のCloudera Hadoop仮想マシンを実装する手順を示します。

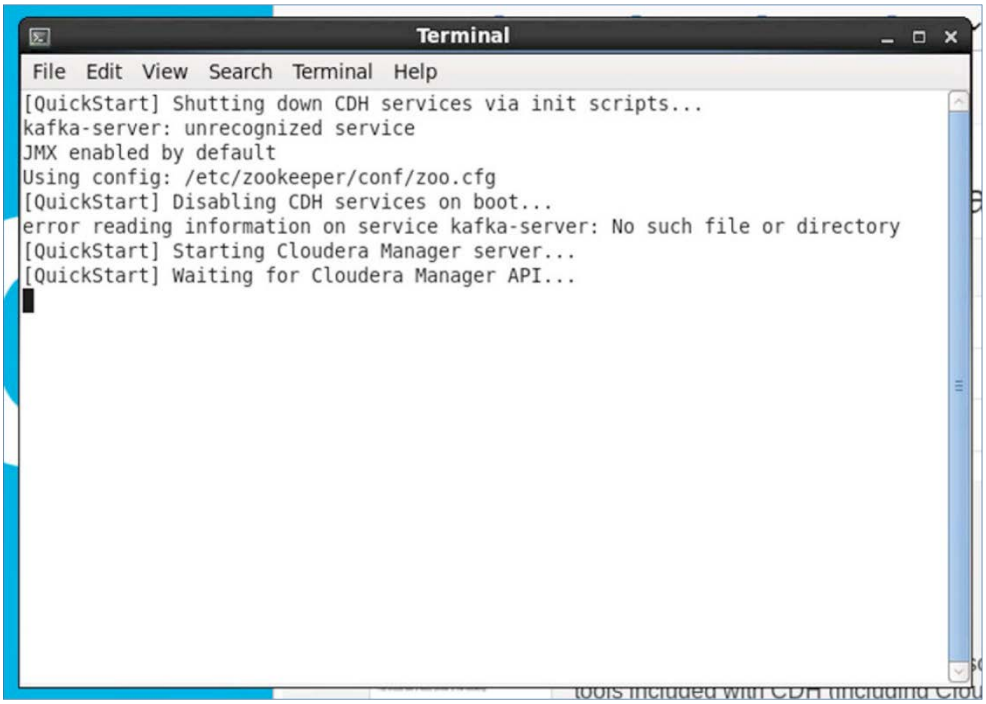
最初に、最新のCloudera HadoopのVMをCloudera.comからダウンロードします。インストールを行い、"Express" バージョンの要件を満たしていることを確認します。このバージョンは、試験的な利用やデモには十分な機能を備えています。"Enterprise" を選択すると、本格的な評価を行うことができますが、必要なRAMやCPUなどのリソース要件も大きくなります。

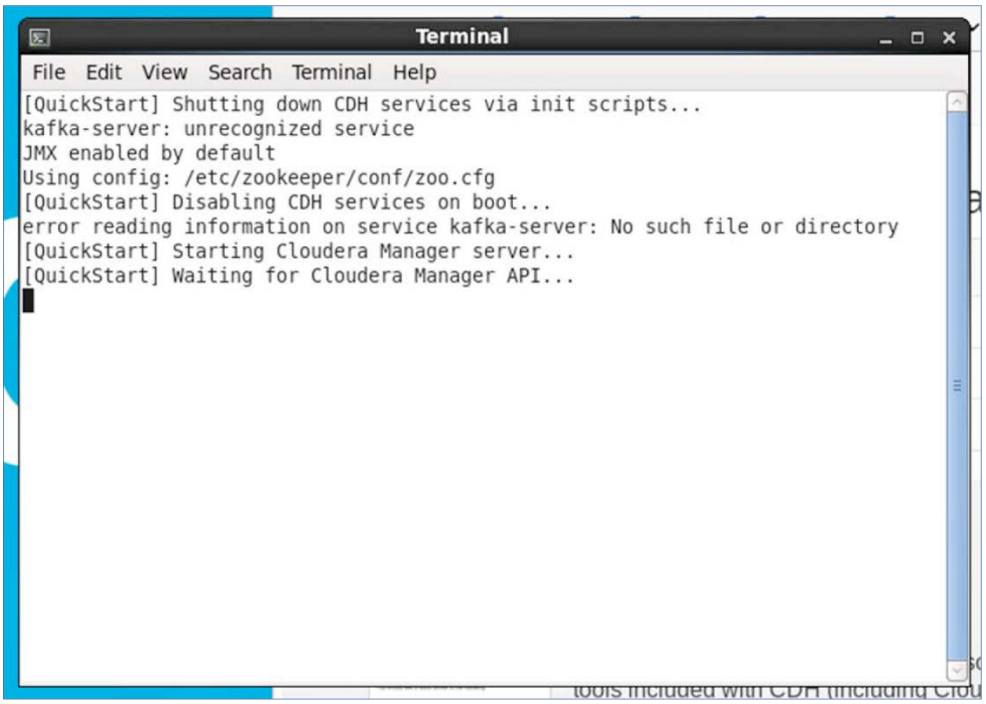
準備ができたら、下のスクリーンショットのように、Expressアイコンをダブルクリックします。一連のスクリプトのインストールが開始され、VMの準備が整います。



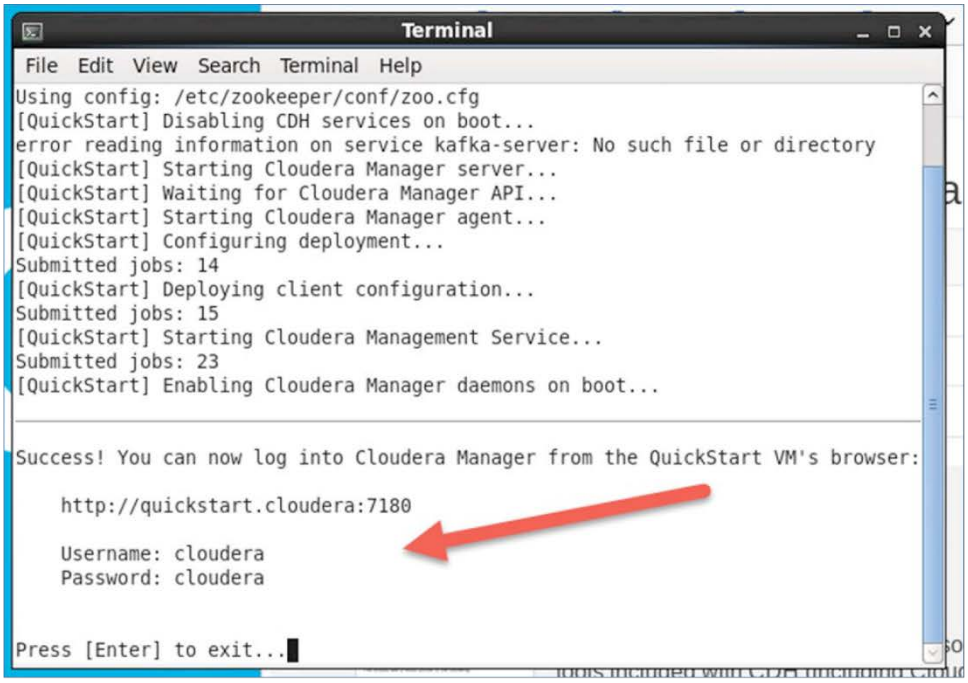


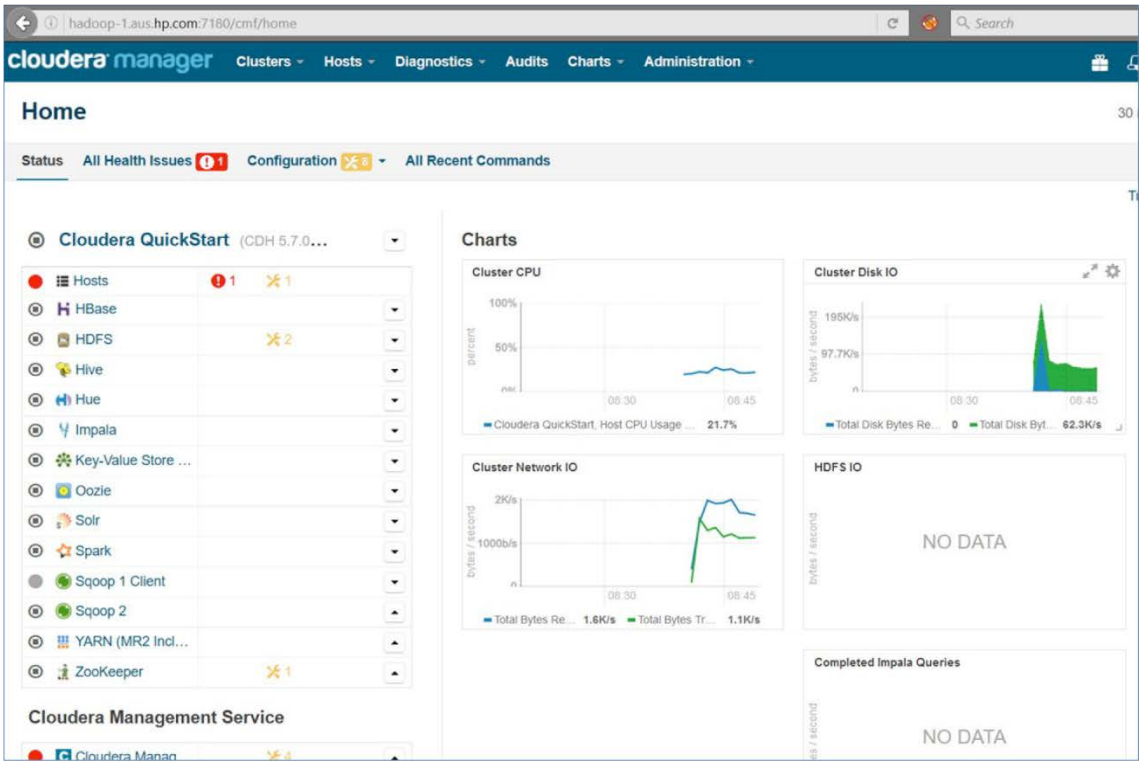
マシンによっては、この作業に数分かかることがあります。





完了すると、下のスクリーンショットのように、Managerに接続するための手順が表示されます。

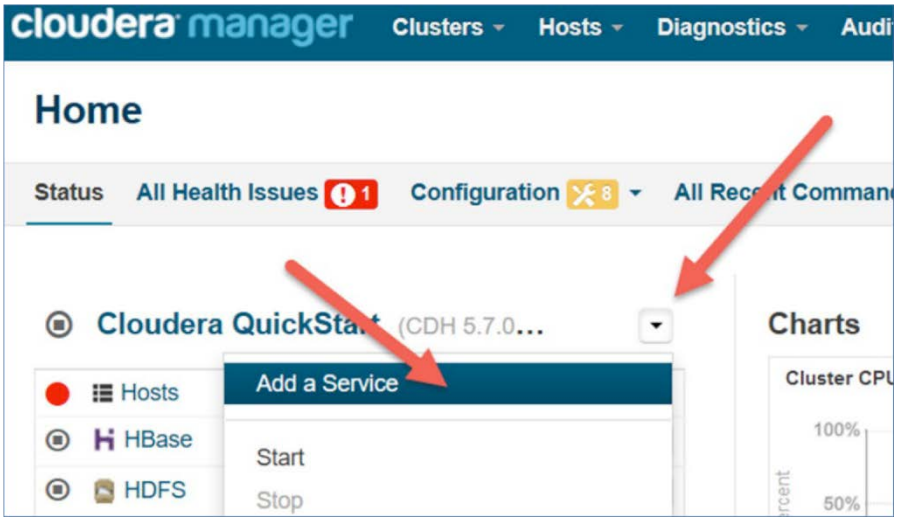




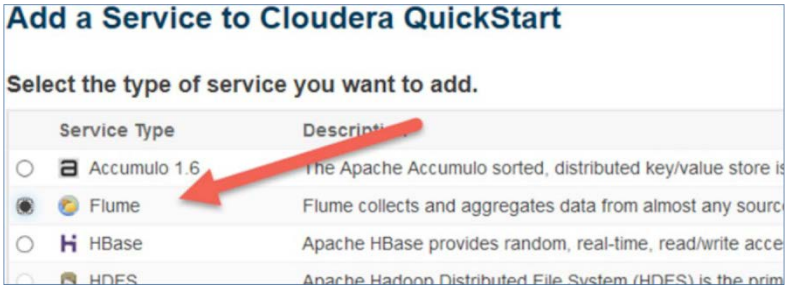
イベントストレージ用のHDFS、ビジュアリゼーションおよびクエリ用のHUE、およびEvent BrokerからのイベントをHDFSに取り込むためのFLUMEなどの、複数のサービスをセットアップする必要があります。以下に、これらのサービスをセットアップする手順を示します。

FLUME

最初にFlumeのインストールを行いますが、ここではまだFlumeの構成は行いません。サービスのインストールのみを行います。

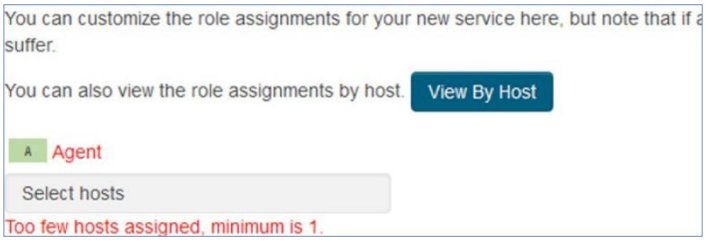


Cloudera QuickStartで下向き矢印をクリックして、[Add a Service] を選択します。

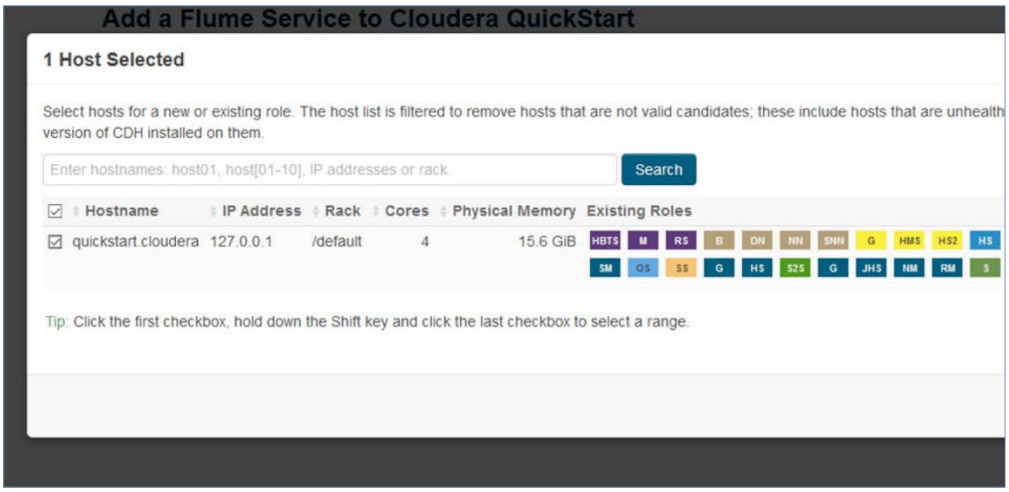


サービスとして [Flume] を選択します

Flumeの依存関係は、HDFSとZooKeeperになります。該当するオプションを選択します。これにより、作業量が最小限に抑えられます。この時点では、HBaseとSolrは必要ありません。



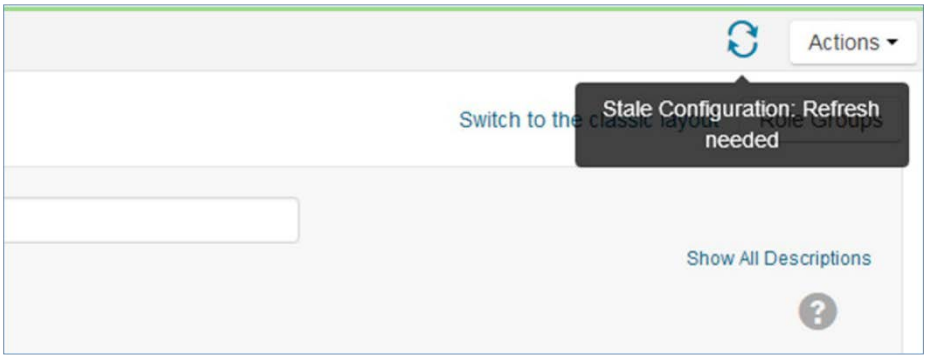
[Select hosts] をクリックし、スクリーンショットのとおりに、使用するVMを選択します。



[Finish] をクリックして、この構成を保存します。

次に、Flumeサービスの構成に進みます。ここでは、ソース/シンクの構成を入力して、Event Brokerのイベントをサブスクライブし、これらをHDFSにパブリッシュできるようにします。

1. [Flume] > [Configuration] をクリックします。
2. [Configuration File] までスクロールして、すべての内容を削除し、以下に示す内容に置き換えます。
3. 構成を保存してサービスを再開するか、「クラスターを更新」します。どちらにしても、構成ファイルをリロードする必要があります。




```
###############################################################################
#Working Flume/Kafka configuration file for Cloudera / Hadoop / Event Broker 1.0
#######################################################
#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.zookeeperConnect = eb-1.aus.hp.com:2181
tier1.sources.source1.topic = BlueCoatEvents
tier1.sources.source1.groupId = 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:8020/user/cloudera/flume/events/%y/%m
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
```

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

以下に示すように、行の一部を更新する必要があります。

```
tier1.sources.source1.zookeeperConnect = eb-1.aus.hp.com:2181
```

```
Your Event Broker node #1—with ZK port (not KAFKA port)
tier1.sources.source1.topic = BlueCoatEvents
```

最初のトピックです。

```
tier1.sources.source1.groupId = flume
```

これは、それぞれのコンシューマーグループ名です。

```
tier1.sinks.sink1.hdfs.rollInterval = 360
```

HDFSでイベントファイルをロールオーバーする頻度です。この値は、想定される取り込み速度や固有のニーズに基づいて調整する必要があります。デモの場合は360で問題ありません。

```
tier1.sinks.sink1.hdfs.filePrefix = cefEvents
tier1.sinks.sink1.hdfs.fileSuffix = .cef
```

上記は、HDFSのファイルプレフィックスとサフィックスです。ニーズに合わせて変更します。

```
tier1.sinks.sink1.hdfs.path = hdfs://localhost:8020/user/cloudera/flume/events/%y/%m
```

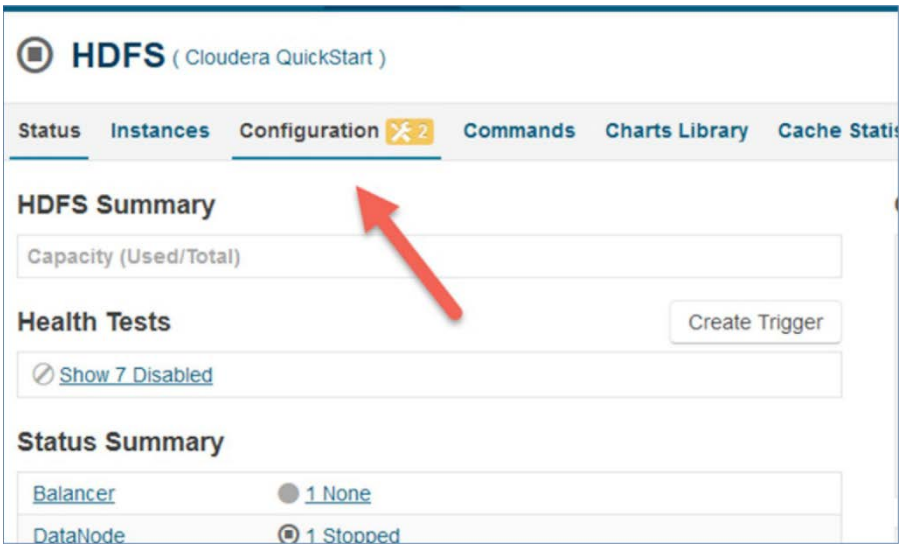
これはシンクで、HDFSのこの場所にファイルが書き込まれます。ここでは、Clouderaユーザーのホームディレクトリに、flume/events/のディレクトリ構造を追加し、末尾に年/月を追加した場所にファイルを配置するように指定しています。これにより、月と年でファイルを分けることができます。これは、それぞれの好みに合わせて設定します。

Flumeはユーザー**Flume**として実行され、通常は**/user/cloudera/**への書き込みアクセス権を持っていないため、アクセス権限を**o+w**に変更する必要があります。これはベストプラクティスではありませんが、デモ用にファイルを書き込めるようにしています。**WebGUI**でこれを行う方法については、以下の**HUE**の構成を参照してください。

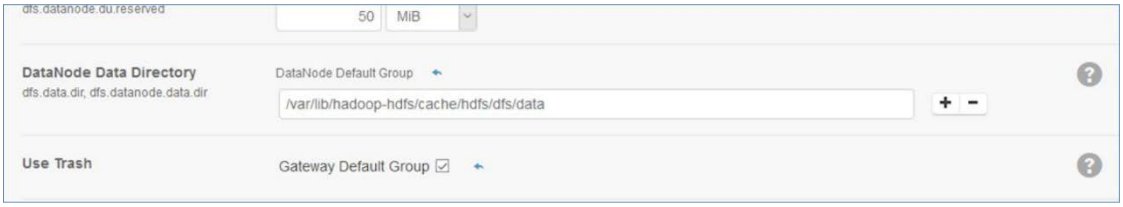
HDFS

ManagerでHDFSをクリックします。

[Configuration] をクリックします。



この構成では、イベントデータ用に複数の場所を指定できることに注意してください。デフォルトで、VMで提供される領域はわずかであるため、長期間使用する必要がある場合は、仮想ディスクの追加を検討します。ディスクがいっぱいになると、一部のサービスが正しく動作しなくなります。



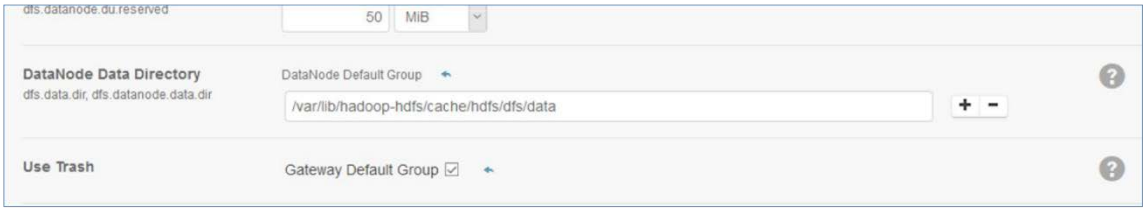
ここで変更した内容を保存します。その他の項目はすべてデフォルトのままになります。

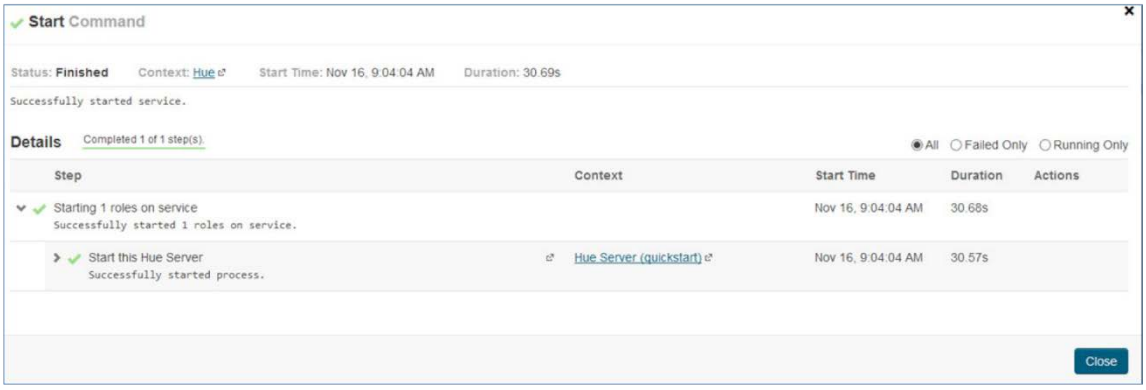
下のスクリーンショットのように、HDFSサービスを開始します。

HUE

次に、HUEを構成します。これにより、HDFSからのイベントを容易に確認できるようになります。これはビジュアリゼーションに役立つわけではありませんが、少なくともイベントを全体としてキャプチャーする機能を示すことはできます。

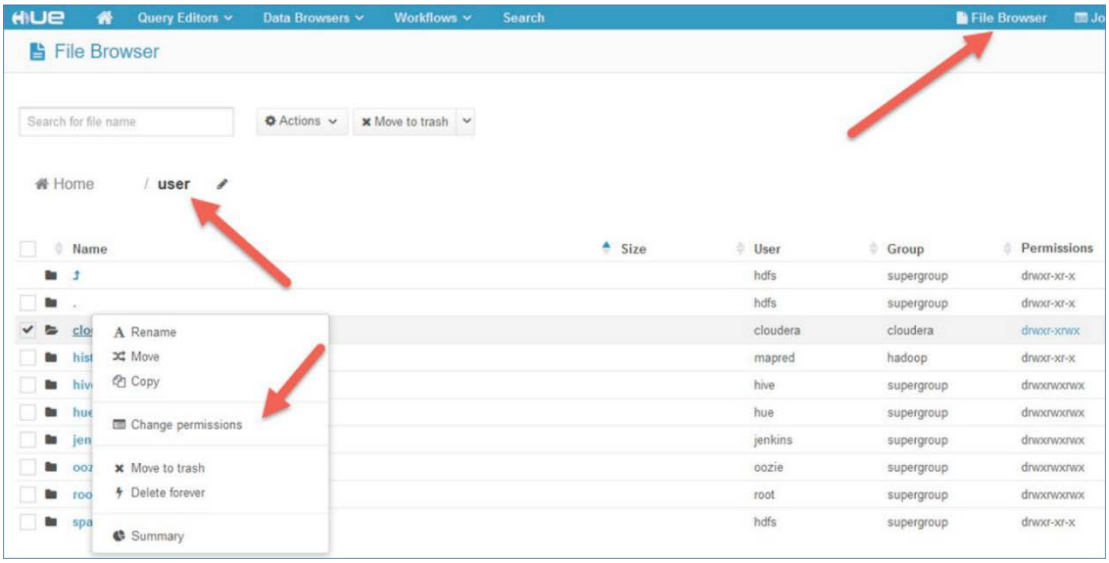
ManagerでHUEサービスを開始します。

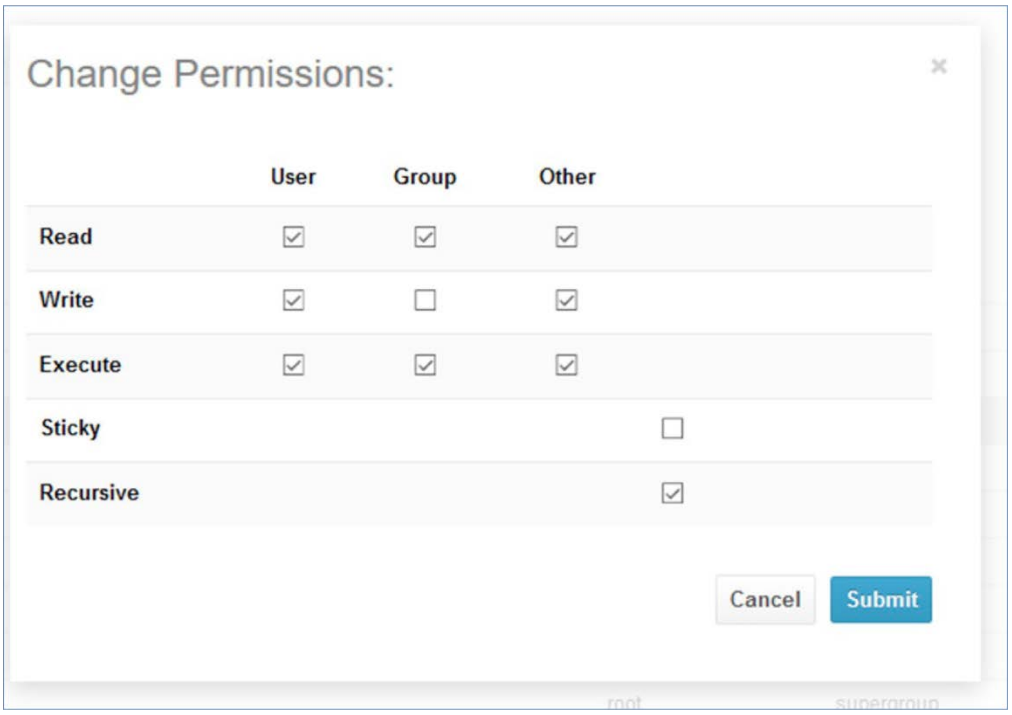




サービスが開始されたら、HTTPでポート8888 (デフォルト) を使用してHUEに接続できます。ユーザーclouderaを使って再度ログインします。ウィザードをスキップします (ウィザードは必要ありません)。

1. 上部メニューの [File Browser] をクリックします。
2. /user/を参照します。
3. “cloudera” を右クリックします。
4. [Change permission] を選択します。
5. “Other” に書き込みアクセスを付与し、[Recursive] にチェックを入れます。これで、FlumeがHDFSにファイルを書き込めるようになります。

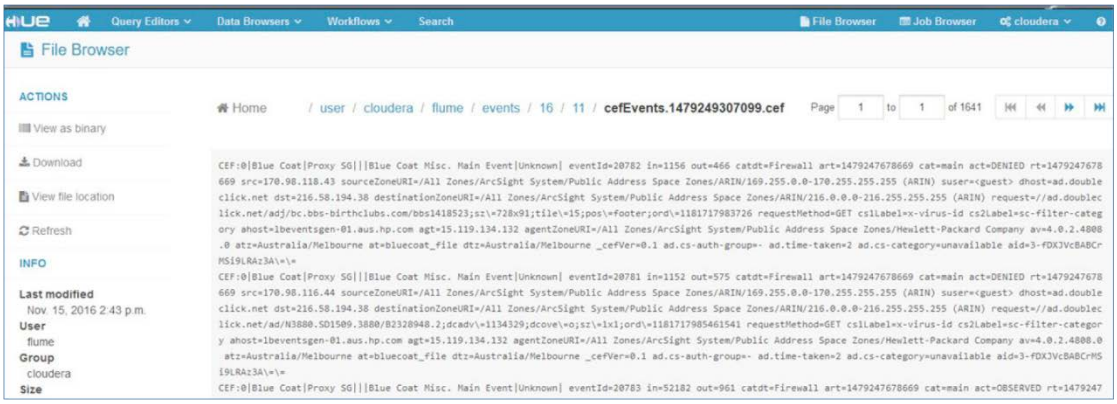




FlumeおよびHDFSのサービスがまだ開始されていない場合は、これらのサービスを開始する必要があります。これらのサービスが開始されると、下のスクリーンショットに示すように、"flume" というディレクトリが/user/cloudera/の下に表示されます。

所有者がユーザーflumeであることに注意してください。これは想定どおりです。ここで示す手順は、稼働モデルを実現するのに必要な最小限の手順です。

次に、ディレクトリを保管されている.cefファイルまで移動します。ここで、ファイルを選択するとファイルが開いて、ファイルのCEFの内容が表示されます。この時点では、イベントの取り込みが継続して行われるため、VMがイベントでいっぱいになります。これを止めるため、ManagerでFlumeサービスを停止します。



まとめ

セキュリティアナリストは、ネットワークに接続された利用可能なすべてのデータを使用して、脅威を検出して対応しようと奮闘しています。ArcSight Data Platformは、オープンアーキテクチャーを活用することで、境界のないデータの可視化を実現し、さまざまなユースケースでの利用を可能にします。また、リアルタイムでデータをエンリッチ化し、任意の場所から任意の場所へのデータを移動できるようにすることで、脅威の検出、調査、および対応を効果的に行えるようにします。ArcSight Data Platformは、大規模な環境で必要とされる信頼性の高いオープンアーキテクチャーのデータ収集プラットフォームを提供することで、インテリジェントなセキュリティオペレーションの基盤を構築します。

リソースおよび関連リンク

ArcSight Data Platform: www.microfocus.com/adp

Open Referral: openreferral.org/faq

Micro Focus Software: software.microfocus.com

Micro Focus Services: software.microfocus.com

Micro Focusサポートおよびドライバー: microfocus.com/support-and-services/

詳細情報

www.microfocus.com/adp

Micro Focusについて

Micro Focusは、40年の実績を持つグローバルなソフトウェア企業で、お客様がリスクを抑えながら迅速に技術革新を行うためのエンタープライズソフトウェアソリューションの提供とサポートを行っています。Micro Focusは、ソフトウェアとセキュリティにおける実績のあるノウハウを活かすことで、お客様が既存のクリティカルなITインフラストラクチャーやビジネスアプリケーションを最大限に活用しながら、最新のテクノロジーソリューションを利用できるように支援しています。これにより、お客様は既存のビジネスロジックやアプリケーションに最新のテクノロジーを結合したITシステムの構築、運用、およびセキュリティ保護を行い、ますます複雑化するビジネスニーズに対応することができます。

Micro Focusについて詳しくは、こちらのビデオをご覧ください。

Micro Focusの商標情報

MICRO FOCUSおよびMicro Focusロゴは、英国、米国、およびその他の国におけるMicro Focus (IP) Limitedまたはその子会社の商標または登録商標です。その他の商標については、それぞれの所有者が権利を有しています。

企業情報

会社名: Micro Focus International plc

登記場所: イングランドおよびウェールズ

登記番号: 5134647

登記住所: The Lawn, 22-30 Old Bath Road, Berkshire, RG14 1Q

マイクロフォーカスエンタープライズ株式会社
0120 923 333

www.microfocus-enterprise.co.jp