AWS 白皮书

AWS 上的流数据解决方案（使用 Amazon Kinesis）

AWS 上的流数据解决方案（使用 Amazon Kinesis）: AWS 白皮书

Table of Contents

摘要 1
摘要 1
简介 2
实时和近实时应用场景 2
批处理和流式处理之间的区别 2
流式处理挑战 3
流式传输数据解决方案：示例 4
场景 1：基于位置的互联网服务 4
Amazon Kinesis Data Streams 4
使用 AWS Lambda 处理数据流 6
总结 6
场景 2：为安全团队提供近实时数据 6
Amazon Kinesis Data Firehose 7
总结 11
场景 3：为数据洞察流程准备点击流数据 12
AWS Glue 和 AWS Glue 流式处理 13
Amazon DynamoDB 14
Amazon SageMaker 和 Amazon SageMaker 服务终端节点 14
实时推理数据洞察 15
总结 15
场景 4：设备传感器实时异常检测和通知 15
Amazon Kinesis Data Analytics 16
适用于 Apache Flink 应用程序的 Amazon Kinesis Data Analytics 17
场景 5：使用 Apache Kafka 进行实时遥测数据监控 19
Amazon Managed Streaming for Apache Kafka (Amazon MSK) 20
迁移到 Amazon MSK 21
结论和贡献者 24
结论 24
贡献者 24
文档修订 25

AWS 上的流数据解决方案

发布日期：2021 年 9 月 1 日（文档修订）

摘要

数据工程师、数据分析师和大数据开发人员正在寻求将其分析从批处理分析演变为实时分析，以便公司能够了解其客户、应用程序和产品目前正在做什么，并迅速做出反应。本白皮书讨论从批处理分析到实时分析的演进。它介绍如何使用 Amazon Kinesis Data Streams、Amazon Kinesis Data Firehose、Amazon EMR、Amazon Kinesis Data Analytics、Amazon Managed Streaming for Apache Kafka (Amazon MSK) 以及其他服务来实现实时应用程序，并提供使用这些服务的常见设计模式。

简介

由于不断生成数据流的数据源的爆炸性增长，如今的企业以巨大的规模和极高的速度接收数据。无论是来自应用程序服务器的日志数据、来自网站和移动应用程序的点击流数据，还是来自物联网 (IoT) 设备的遥测数据，所包含的信息都可以帮助您了解客户、应用程序和产品目前正在做什么。

能够实时处理和分析此类数据对于实现如下目标至关重要：持续监控应用程序以确保长久的服务正常运行时间；实现促销优惠和产品建议个性化。实时和近实时处理还可以提高其他常见使用案例（如网站分析和机器学习）的准确性和可行性，因为它可以在几秒或几分钟内（而不是数小时或数天内）将数据提供给相关应用程序。

实时和近实时应用场景

您可以将流式传输数据服务用于实时和近实时应用，例如应用程序监控、欺诈侦测和直播排行榜。实时使用案例要求毫秒级的端到端延迟 - 从摄入到处理，直至将结果发送到目标数据存储和其他系统。例如，Netflix 使用 Amazon Kinesis Data Streams 监控其所有应用程序之间的通信，从而快速发现和解决问题，让客户享受到正常运行时间很长、可用性很高的服务。虽然最常用的使用案例是应用程序性能监控，但归属此类别的广告技术、游戏和 IoT 领域的实时应用越来越多。

常见的近实时使用案例包括针对数据科学和机器学习 (ML) 的数据存储进行分析。您可以使用流式传输数据解决方案将实时数据持续加载到数据湖中。然后，当有新数据可用时，您可以更频繁地更新机器学习 (ML) 模型，确保结果的准确性和可靠性。例如，Zillow 使用 Kinesis Data Streams 采集公有记录数据和多重挂牌服务系统 (MLS) 报价，然后近实时地向房屋买卖双方提供最新的住宅估价信息。ZipRecruiter 将 Amazon MSK 用于其事件日志记录管道，该管道是一个重要的基础设施组件，每天从 ZipRecruiter 求职平台收集、存储并持续处理超过六十亿个事件。

批处理和流式处理之间的区别

与传统上用于批处理分析的工具相比，您需要一套不同的工具来收集、准备和处理实时流式传输数据。使用传统的分析，您收集数据，定期将其加载到数据库中，然后在数小时、数天或数周后对其进行分析。分析实时数据需要采用不同的方法。流式处理应用程序实时、连续处理数据，甚至在存储数据之前也是如此。流式传输数据可能以极快的速度进入，数据量可能随时上下变化。流式数据处理平台必须能够处理传入数据的速度和可变性，并在数据到达时对其进行处理，通常每小时处理数百万到数亿个事件。

流式处理挑战

与传统数据分析技术相比，在实时数据到达时对其进行处理，您做出决策的速度会快得多。但是，构建和运行自己的自定义流式数据管道很复杂且占用大量资源：

- 您必须构建这样一个系统，它能够经济高效地收集、准备和传输来自数千个数据源的数据。
- 您需要微调存储和计算资源，以便高效地对数据进行批处理和传输，从而实现高吞吐量和低延迟。
- 您必须部署和管理服务器机群才能扩展系统，以便能够处理将涌向它的不同速度的数据。

版本升级是一个复杂且成本高昂的过程。构建此平台后，您必须监控系统，并从任何服务器或网络故障中恢复（通过从流中的适当点赶上数据处理来实现），而不创建重复的数据。您还需要专门的基础设施管理团队。所有这些都需要花费宝贵的时间和金钱，最终，大多数公司根本无法实现这一目标，而是必须适应现状，利用已存在数小时或数天之久的信息来运营其业务。

流式传输数据解决方案：示例

场景 1：基于位置的互联网服务

InternetProvider 公司为世界各地的用户提供具有各种带宽选项的互联网服务。当用户注册互联网时，InternetProvider 公司会根据用户的地理位置为用户提供不同的带宽选项。鉴于这些要求，InternetProvider 公司实施了 Amazon Kinesis Data Streams 来使用用户详细信息和位置。在将用户详细信息和位置发布回应用程序之前，用不同的带宽选项丰富这些信息。AWS Lambda 支持这种实时丰富措施。

使用 AWS Lambda 处理数据流

Amazon Kinesis Data Streams

借助 Amazon Kinesis Data Streams，您可以使用常用的流式处理框架构建自定义的实时应用程序，并将流式传输数据加载到许多不同的数据存储中。一个 Kinesis 流可以配置为持续接收来自数十万个数据生成者的事件，这些数据生成者来自网站点击流、IoT 传感器、社交媒体源和应用程序日志等来源。几毫秒内，数据即可供应用程序进行读取和处理。

使用 Kinesis Data Streams 实施解决方案时，您可以创建称为 Kinesis Data Streams 应用程序的自定义数据处理应用程序。典型的 Kinesis Data Streams 应用程序将来自 Kinesis 流的数据作为数据记录读取。

确保放入 Kinesis Data Streams 的数据具有高可用性和弹性，几毫秒内即可使用。您可将来自数千个来源的点击流、应用程序日志和社交媒体等各种类型的数据持续添加到 Kinesis 流。在数秒内，Kinesis 应用程序便可以从流中读取和处理数据。

Amazon Kinesis Data Streams 是一项完全托管的流式传输数据服务。它管理在数据吞吐量层面流式处理您的数据所需的基础设施、存储、联网和配置。

将数据发送到 Amazon Kinesis Data Streams

可以通过多种方法将数据发送到 Kinesis Data Streams，从而为解决方案的设计提供灵活性。

- 您可以利用多种常用语言支持的 AWS 软件开发工具包之一来编写代码。
- 您可以使用 Amazon Kinesis 代理，这是一款用于将数据发送到 Kinesis Data Streams 的工具。

Amazon Kinesis Producer Library (KPL) 使开发人员能够对一个或多个 Kinesis 数据流实现较高的写入吞吐量，从而简化生成者应用程序的开发过程。

KPL 是一个易于使用、高度可配置的库，您可以将其安装在主机上。它在您的生成者应用程序代码和 Kinesis Streams API 操作之间充当中介。有关 KPL 及其以同步和异步方式生成事件的功能以及代码示例的更多信息，请参阅使用 KPL 写入 Kinesis Data Streams

Kinesis Data Streams API 中有两个不同的操作可向流添加数据：`PutRecords` 和 `PutRecord`。`PutRecords` 操作对于每个 HTTP 请求向您的流发送多条记录，而 `PutRecord` 对于每个 HTTP 请求提交一条记录。要为大多数应用程序实现更高的吞吐量，请使用 `PutRecords`。

有关这些 API 的更多信息，请参阅向流添加数据。每个 API 操作的详细信息可在 Amazon Kinesis Data Streams API 参考中找到。

在 Amazon Kinesis Data Streams 中处理数据

要读取和处理来自 Kinesis 流的数据，您需要创建一个使用者应用程序。可以通过多种方法为 Kinesis Data Streams 创建使用者。其中一些方法包括使用 Amazon Kinesis Data Analytics，通过 KCL、AWS Lambda、AWS Glue 流式处理 ETL 任务以及直接使用 Kinesis Data Streams API 来分析流数据。

可以使用 KCL 开发适用于 Kinesis Data Streams 的使用者应用程序，KCL 可以帮助您使用和处理来自 Kinesis Data Streams 的数据。KCL 负责许多与分布式计算相关的复杂任务，例如对多个实例实现负载均衡、对实例故障做出响应、对已处理的数据执行检查点操作以及对重新分片做出应对。KCL 可让您将精力放在编写记录处理逻辑方面。有关如何构建您自己的 KCL 应用程序的更多信息，请参阅使用 Kinesis 客户端库。

您可以订阅 Lambda 函数，以自动从您的 Kinesis 流中读取批量记录，并在流中检测到记录时对其进行处理。AWS Lambda 定期轮询流（每秒一次）以查找新记录，当它检测到新记录时，它会调用 Lambda 函数，同时将新记录作为参数传递。Lambda 函数仅在检测到新记录时才运行。您可以将 Lambda 函数映射到共享吞吐量使用者（标准迭代器）。

当您需要专用吞吐量而又不想与从流中接收数据的其他使用者争用时，可以构建使用增强扇出功能的使用者。利用此功能，使用者可以从流中接收记录，其数据吞吐量高达每分片 2 MB/秒。

在大多数情况下，应使用 Kinesis Data Analytics、KCL、AWS Glue 或 AWS Lambda 来处理流中的数据。但是，如果您愿意，可以使用 Kinesis Data Streams API 从头开始创建使用者应用程序。Kinesis Data Streams API 提供了用于从流检索数据的 `GetShardIterator` 和 `GetRecords` 方法。

在此拉取模型中，您的代码直接从流的分片中提取数据。有关使用 API 编写自己的使用者应用程序的更多信息，请参阅使用适用于 Java 的 AWS 软件开发工具包开发具有共享吞吐量的自定义使用者。有关此 API 的详细信息可在 Amazon Kinesis Data Streams API 参考中找到。

使用 AWS Lambda 处理数据流

AWS Lambda 使您无需预置或管理服务器即可运行代码。借助 Lambda，您可以为几乎任何类型的应用程序或后端服务运行代码，而且无需任何管理。您只需上载代码，Lambda 就会处理运行和扩展具有高度可用性的代码所需的一切工作。您可以将您的代码设置为自动从其他 AWS 服务触发，或者直接从任何 Web 或移动应用程序调用。

AWS Lambda 与 Amazon Kinesis Data Streams 原生集成。使用此原生集成时，将抽象化处理轮询操作、检查点操作以及错误处理的复杂性。这使得 Lambda 函数代码能够专注于业务逻辑处理。

您可以将 Lambda 函数映射到共享吞吐量使用者（标准迭代器）或具有增强扇出功能的专用吞吐量使用者。对于标准迭代器，Lambda 使用 HTTP 协议轮询 Kinesis 流中的每个分片以查找记录。为了最大限度地减少延迟并最大限度地提高读取吞吐量，您可以创建具有增强扇出功能的数据流使用者。此架构中的流使用者可以获得与每个分片的专用连接，而无需与从同一流中读取的其他应用程序竞争。Amazon Kinesis Data Streams 通过 HTTP/2 将记录推送到 Lambda。

原定设置情况下，只要流中有记录，AWS Lambda 就会调用您的函数。要为批处理场景缓冲记录，您可以在事件源处实施多达五分钟的批处理时段。如果您的函数返回一个错误，则 Lambda 将重试批处理，直到处理成功或数据过期。

总结

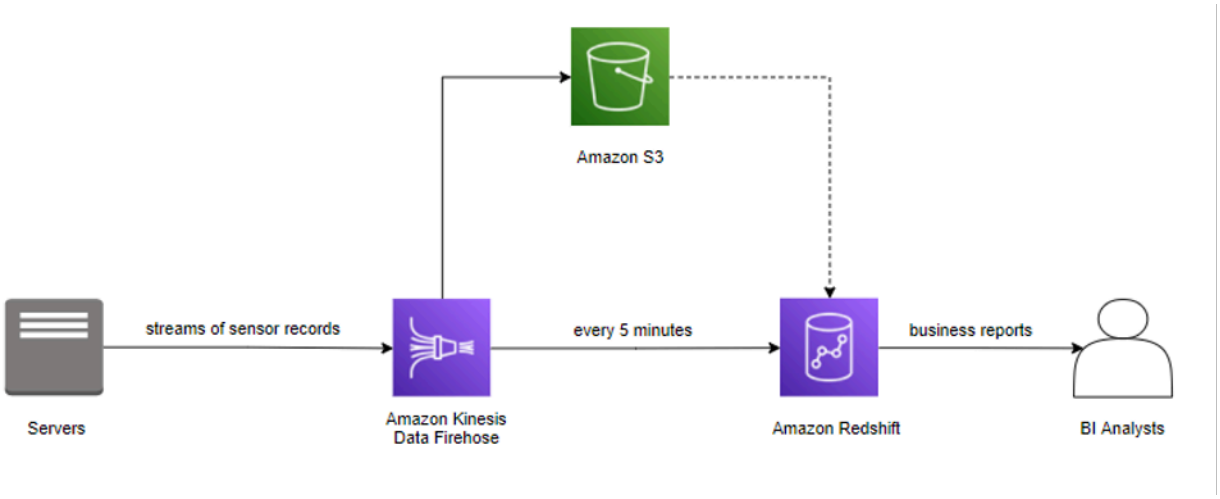
InternetProvider 公司已利用 Amazon Kinesis Data Streams 来流式传输用户详细信息和位置。AWS Lambda 使用记录流，通过存储在函数库中的带宽选项来丰富数据。丰富数据后，AWS Lambda 将带宽选项发布回应用程序。Amazon Kinesis Data Streams 和 AWS Lambda 负责服务器的预置和管理，使 InternetProvider 公司能够更加专注于业务应用程序开发。

场景 2：为安全团队提供近实时数据

ABC2Badge 公司为企业或大型活动（例如 AWS re:Invent）提供传感器和徽章。用户注册参加此活动，并收到传感器在园区内获取的独特徽章。当用户经过传感器时，他们的匿名信息会被被记录到关系数据库中。

在即将举行的活动中，由于与会者人数众多，活动安全团队已要求 ABC2Badge 每 15 分钟收集一次园区中人员最密集区域的数据。这将使安全团队有足够的时间做出反应，并按比例将安保人员分散到各个人员密集区域。鉴于安全团队提出的这一新要求以及构建流解决方案的经验不足，为了近实时地处理数据，ABC2Badge 正在寻找一种简单但可扩展且可靠的解决方案。

他们目前的数据仓库解决方案是 Amazon Redshift。在查看 Amazon Kinesis 服务的功能时，他们认识到 Amazon Kinesis Data Firehose 可以接收数据记录流，根据缓冲区大小和/或时间间隔对记录进行批处理，然后将其插入到 Amazon Redshift 中。他们创建了一个 Kinesis Data Firehose 传输流并对其进行了配置，以便每五分钟将数据复制到他们的 Amazon Redshift 表中。作为这一新解决方案的一部分，他们在服务器上使用了 Amazon Kinesis 代理。每五分钟，Kinesis Data Firehose 就会将数据加载到 Amazon Redshift 中，其中，商业智能 (BI) 团队可以执行数据分析并每隔 15 分钟向安全团队发送一次数据。



使用 Amazon Kinesis Data Firehose 的新解决方案

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose 是将流数据加载到 AWS 的最简单方式。它可以捕获、转换流数据并将其加载到 Amazon Kinesis Data Analytics、Amazon Simple Storage Service (Amazon S3)、Amazon Redshift、Amazon OpenSearch Service (OpenSearch Service) 和 Splunk 中。此外，Kinesis Data Firehose 可以将流数据加载到任何自定义 HTTP 终端节点或由受支持的第三方服务提供商拥有的 HTTP 终端节点中。

Kinesis Data Firehose 旨在与您目前已在使用的现有商业智能工具和控制面板配合，从而实现近实时的分析。这是一项完全托管式无服务器服务，可以自动扩展以匹配数据吞吐量，并且无需持续管理。Kinesis Data Firehose 可以在加载数据前对数据进行批处理、压缩和加密，从而最大程度地减少在目标位置占用的存储量，同时提高安全性。它还可以使用 AWS Lambda 转换源数据，并将转换后的

数据传送到目标位置。您可以配置数据生成者向 Kinesis Data Firehose 发送数据，然后 Kinesis Data Firehose 将数据自动传输到您指定的目标位置。

将数据发送到 Firehose 传输流

要将数据发送到您的传输流，有几种选项。AWS 为许多常用的编程语言提供了软件开发工具包，每个软件开发工具包都为 Amazon Kinesis Data Firehose 提供了 API。AWS 有一个实用程序可帮助将数据发送到您的传输流。Kinesis Data Firehose 已与其他 AWS 服务集成，可将这些服务中的数据直接发送到您的传输流。

使用 Amazon Kinesis 代理

Amazon Kinesis 代理是一个独立的软件应用程序，它持续监控一组日志文件，以查找要发送到传输流的新数据。此代理会自动处理文件轮换、检查点操作、出现故障时的重试，并发出 Amazon CloudWatch 指标以监控传输流并排除其故障。可以将其他配置（例如数据预处理、监控多个文件目录以及写入多个传输流）应用于此代理。

此代理可以安装在基于 Linux 或 Windows 的服务器上，例如 Web 服务器、日志服务器和数据库服务器。安装此代理后，只需指定它将监控的日志文件及其将发送到的传输流即可。此代理将持久、可靠地将新数据发送到传输流。

将 API 与 AWS 软件开发工具包和 AWS 服务一起用作源

Kinesis Data Firehose API 提供两种向传输流发送数据的操作：`PutRecord` 在一次调用中发送一条数据记录。`PutRecordBatch` 在一次调用中发送多条数据记录，并且对于每个生成者可以实现更高的吞吐量。对于每种方法，使用此方法时都必须指定传输流的名称和数据记录或数据记录数组。有关 Kinesis Data Firehose API 操作的更多信息和示例代码，请参阅使用 AWS 软件开发工具包写入 Firehose 传输流。

Kinesis Data Firehose 还可以与 Kinesis Data Firehose、CloudWatch Logs、CloudWatch Events、Amazon Simple Notification Service (Amazon SNS)、Amazon API Gateway 和 AWS IoT 一起运行。您可以通过可扩展且可靠的方式将数据流、日志、事件和 IoT 数据直接发送到 Kinesis data Firehose 目标位置。

在传输到目标位置之前处理数据

在某些情况下，您可能希望在将流数据传输到其目标位置之前对其进行转换或增强。例如，数据生成者可能会在每条数据记录中发送非结构化文本，而您需要先将其转换为 JSON，然后再将其传输到 OpenSearch Service。或者，在将数据存储在 Amazon S3 中之前，您可能希望将 JSON 数据转换为一种列式文件格式，如 Apache Parquet 或 Apache ORC 。

Kinesis Data Firehose 具有内置的数据格式转换功能。使用此功能，可以轻松地将 JSON 数据流转换为 Apache Parquet 或 Apache ORC 文件格式。

数据转换流

为了启用流数据转换，Kinesis Data Firehose 使用您创建的 Lambda 函数来转换数据。Kinesis Data Firehose 将传入的数据缓冲到函数的指定缓冲区大小，然后以异步方式调用指定的 Lambda 函数。转换后的数据将从 Lambda 发送到 Kinesis Data Firehose，然后 Kinesis Data Firehose 将数据传输到目标位置。

数据格式转换

还可以启用 Kinesis Data Firehose 数据格式转换，这会将 JSON 数据流转换为 Apache Parquet 或 Apache ORC。此功能只能将 JSON 转换为 Apache Parquet 或 Apache ORC。如果您有 CSV 格式的数据，则可以通过 Lambda 函数将该数据转换为 JSON，然后应用数据格式转换。

数据传输

作为近实时的传输流，Kinesis Data Firehose 会缓冲传入的数据。达到传输流的缓冲阈值后，数据将传输到您配置的目标位置。Kinesis Data Firehose 将数据传输到每个目标位置的方式存在一些差异，本文将在以下各节中对此进行考察。

Amazon S3

Amazon S3 是一种对象存储，具有简单的 Web 服务接口，可用于在 Web 上的任何位置存储和检索任意数量的数据。它旨在提供 99.999999999% 的持久性，并且可以在全球范围内大规模传递数万亿对象。

将数据传输到 Amazon S3

为了将数据传输到 Amazon S3，Kinesis Data Firehose 根据传输流的缓冲配置串联多个传入记录，然后将它们作为一个 S3 对象传输到 Amazon S3。向 S3 传输数据的频率由 S3 缓冲区大小（1 MB 到 128 MB）或缓冲区间隔（60 秒到 900 秒）决定，以先到者为准。

向 S3 存储桶传输数据可能会由于各种原因而失败。例如，存储桶不再存在、Kinesis Data Firehose 代入的 AWS Identity and Access Management (IAM) 角色没有访问存储桶的权限。在此类情况下，Kinesis Data Firehose 会持续重试长达 24 小时，直到传输成功为止。Kinesis Data Firehose 的最长数据存储时间为 24 小时。如果数据传输失败超过 24 小时，数据将丢失。

Amazon Redshift

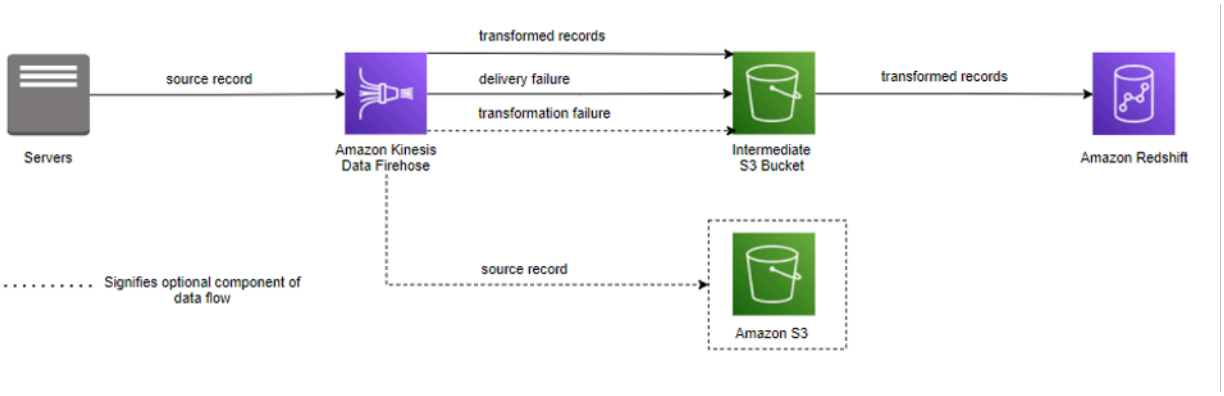
Amazon Redshift 是一种快速、完全托管式数据仓库，使用标准 SQL 和您现有的 BI 工具分析所有数据，简单且经济高效。利用它，您可以使用高性能本地磁盘上的列式存储，通过复杂的查询优化对 PB 级结构化数据运行复杂的分析查询，并能大规模运行并行查询。

将数据传输到 Amazon Redshift

为了将数据传输到 Amazon Redshift，Kinesis Data Firehose 首先以前面描述的格式将传入的数据传输到您的 S3 存储桶。然后，Kinesis Data Firehose 发出 Amazon Redshift COPY 命令，以将数据从 S3 存储桶加载到您的 Amazon Redshift 集群。

从 S3 到 Amazon Redshift 的数据 COPY 操作的频率取决于 Amazon Redshift 集群完成 COPY 命令的速度。对于 Amazon Redshift 目标位置，您可以在创建传输流时指定重试持续时间（0 - 7200 秒），以处理数据传输失败。Kinesis Data Firehose 会在指定的持续时间内重试，如果失败，则跳过该特定批次的 S3 对象。所跳过对象的信息会以清单文件的形式传输到您的 S3 存储桶中的 errors/ 文件夹内，您可以利用该清单文件进行手动回填。

以下是 Kinesis Data Firehose 到 Amazon Redshift 数据流的架构图。尽管此数据流是 Amazon Redshift 独有的，但 Kinesis data Firehose 对于其他目标位置也遵循类似的模式。



从 Kinesis Data Firehose 到 Amazon Redshift 的数据流

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service 是一项完全托管式服务，可以提供各种易于使用的 OpenSearch API 和实时功能，还可以实现生产工作负载需要的可用性、可扩展性和安全性。通过 OpenSearch Service 可以轻松地部署、操作和扩展 OpenSearch，以进行日志分析、全文搜索和应用程序监控。

将数据传输到 OpenSearch Service

为了将数据传输 OpenSearch Service，Kinesis Data Firehose 根据传输流的缓冲配置缓冲传入的记录，然后生成 OpenSearch 批处理请求，以将多条记录编入到 OpenSearch 集群的索引中。向 OpenSearch Service 传输数据的频率由 OpenSearch 缓冲区大小（1 MB 到 100 MB）或缓冲区间隔（60 秒到 900 秒）值决定，以先到者为准。

对于 OpenSearch Service 目标位置，您可以在创建传输流时指定重试时长（0 - 7200 秒）。Kinesis Data Firehose 会重试指定的时长，然后跳过该特定的索引请求。跳过的文档会传输到您的 S3 存储桶中的 `elasticsearch_failed/` 文件夹内，您可以利用它进行手动回填。

Amazon Kinesis Data Firehose 可以基于时间段轮换 OpenSearch Service 索引。根据您选择的轮换选项（`NoRotation`、`OneHour`、`OneDay`、`OneWeek` 或 `OneMonth`），Kinesis Data Firehose 向您指定的索引名称追加协调世界时 (UTC) 到达时间戳的一部分。

自定义 HTTP 终端节点或受支持的第三方服务提供商

Kinesis Data Firehose 可以将数据发送到自定义 HTTP 终端节点或受支持的第三方提供商，例如 Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Splunk 和 Sumo Logic。

自定义 HTTP 终端节点或受支持的第三方服务提供商

为使 Kinesis Data Firehose 能够成功地将数据传输到自定义 HTTP 终端节点，这些终端节点必须使用特定的 Kinesis Data Firehose 请求和响应格式来接受请求和发送响应。

将数据传输到受支持的第三方服务提供商拥有的 HTTP 终端节点时，您可以使用集成的 AWS Lambda 服务创建一个函数，以将传入的记录转换为与服务提供商集成所期望的格式相匹配的格式。

对于数据传输频率，每个服务提供商都有建议的缓冲区大小。请与您的服务提供商联系，了解有关他们建议的缓冲区大小的更多信息。对于数据传输失败的处理，Kinesis data Firehose 首先通过等待来自目标位置的响应来建立与 HTTP 终端节点的连接。Kinesis Data Firehose 会继续建立连接，直到重试持续时间过期。超过该时间之后，Kinesis Data Firehose 会将其视为数据传输失败，并将数据备份到您的 S3 存储桶。

总结

Kinesis Data Firehose 可以持续将您的流数据传输到受支持的目标位置。这是一个完全托管式解决方案，几乎不需要或根本不需要开发。对于 ABC2Badge 公司，使用 Kinesis Data Firehose 是很自然的选择。他们已经在使用 Amazon Redshift 作为其数据仓库解决方案。由于他们的数据源持续写入事

务日志，因此他们能够利用 Amazon Kinesis 代理来流式处理该数据，而无需编写任何其他代码。现在，ABC2Badge 公司已经创建了传感器记录流，并正在通过 Kinesis Data Firehose 接收这些记录，他们可以将其用作安全团队使用案例的基础。

场景 3：为数据洞察流程准备点击流数据

Fast Sneakers 是一家专注于时尚运动鞋的时尚精品店。任何一双鞋的价格都可能会因为库存和潮流趋势的变化而上涨或下降，例如昨晚在电视上发现哪位名人或体育明星穿着某名牌的运动鞋。对于 Fast Sneakers 来说，跟踪和分析这些潮流趋势非常重要，以便可以最大限度地提高收益额。

Fast Sneakers 不希望在项目中引入维护新基础设施方面的额外开销。他们希望能够将开发工作分给适当的各方，其中，数据工程师可以专注于数据转换工作，而他们的数据科学家则可以独立处理其机器学习 (ML) 功能。

为了快速做出响应并根据需求自动调整价格，Fast Sneakers 对重大事件（如点击兴趣和购买数据）进行流式处理，转换和增强事件数据，并将数据提供给机器学习 (ML) 模型。他们的机器学习 (ML) 模型能够确定是否需要调整价格。这使 Fast Sneakers 可以自动修改其定价，以最大限度地提高其产品的利润。

Fast Sneakers 实时价格调整

此架构图显示了利用 Kinesis Data Streams、AWS Glue 和 DynamoDB Streams 创建的 Fast Sneakers 实时流式处理解决方案。通过利用这些服务，他们可以获得具备弹性且可靠的解决方案，而无需花费时间来设置和维护支持基础设施。他们可以通过专注于流提取、转换、加载 (ETL) 任务和机器学习模型，从而将时间花在为公司带来价值的事情上。

为了更好地了解其工作负载中使用的架构和技术，以下是所用服务的一些详细信息。

AWS Glue 和 AWS Glue 流式处理

AWS Glue 是一项完全托管式 ETL 服务，您可以用来登记、清理和丰富数据，并可以在数据存储之间可靠地移动数据。借助 AWS Glue，您可以显著降低创建 ETL 任务所花的成本、复杂性和时间。AWS Glue 是无服务器的，因此无需设置或管理任何基础设施。您仅需为运行任务时所消耗的资源付费。

利用 AWS Glue，您可以使用 AWS Glue 流式处理 ETL 任务创建使用者应用程序。这使您能够利用 Apache Spark 和其他基于 Spark 的模块写入来使用和处理事件数据。本文档的下一部分将更深入地介绍这一场景。

AWS Glue Data Catalog

AWS Glue Data Catalog 包含对以下数据的引用：这些数据在 AWS Glue 中用作 ETL 任务的源和目标。AWS Glue Data Catalog 是数据的位置、架构和运行时指标的索引。您可以使用数据目录中的信息来创建和监控您的 ETL 任务。数据目录中的信息将存储为元数据表，其中每个表指定单一数据存储。通过设置爬网程序，您可以自动评估多种类型的数据存储（包括 DynamoDB、S3 和 Java 数据库连接 (JDBC) 连接的存储），提取元数据和架构，然后在 AWS Glue Data Catalog 中创建表定义。

要在 AWS Glue 流式处理 ETL 任务中使用 Amazon Kinesis Data Streams，最佳实践是在 AWS Glue Data Catalog 数据库的表中定义流。您可以使用 Kinesis 流定义源于流的表，Kinesis 流是支持的多种格式之一（CSV、JSON、ORC、Parquet、Avro 或使用 Grok 的客户格式）。您可以手动输入架构，也可以将此步骤留给 AWS Glue 任务以在任务运行时期间确定。

AWS Glue 流式处理 ETL 任务

AWS Glue 在 Apache Spark 无服务器环境中运行您的 ETL 任务。AWS Glue 在用其自己的服务账户预置和管理的虚拟资源上运行这些任务。除了能够运行基于 Apache Spark 的任务之外，AWS Glue 还可以通过 DynamicFrames 在 Spark 之上提供更高级别的功能。

DynamicFrames 是支持嵌套数据（如结构和数组）的分布式表。每条记录都是自描述的，旨在实现半结构化数据的架构灵活性。`DynamicFrame` 中的记录既包含数据，也包含描述数据的架构。ETL 脚本中同时支持 Apache Spark `DataFrames` 和 `DynamicFrames`，您可以来回转换它们。`DynamicFrames` 提供了一组用于数据清理和 ETL 的高级转换。

通过在 AWS Glue 任务中使用 Spark Streaming，您可以创建持续运行的流式处理 ETL 任务，并使用来自 Amazon Kinesis Data Streams、Apache Kafka 和 Amazon MSK 等流式处理源的数据。这些任务可以清理、合并和转换数据，然后将结果加载到存储（包括 Amazon S3、Amazon DynamoDB 或 JDBC 数据存储）中。

原定设置情况下，AWS Glue 在 100 秒的时段内处理和写出数据。这可以实现数据的高效处理，并允许对在预计时间之后到达的数据执行聚合。您可以通过调整窗口大小来配置窗口大小，以适应响应速度与聚合的准确性。AWS Glue 式处理流任务使用检查点来跟踪已从 Kinesis Data Streams 中读取的数据。有关在 AWS Glue 中创建流式处理 ETL 任务的演练，请参阅在 AWS Glue 中添加流式处理 ETL 任务。

Amazon DynamoDB

Amazon DynamoDB 是一种键值和文档数据库，可在任何规模下提供延迟不到十毫秒的性能。它是一个完全托管式、多区域、多活动的持久数据库，具有适用于 Internet 规模应用程序的内置安全性、备份和恢复以及内存中缓存。DynamoDB 每天可处理超过十万亿个请求，并可支持每秒超过 2000 万个请求的峰值。

DynamoDB Streams 的更改数据捕获

DynamoDB 流是一种有关 DynamoDB 表中的项目更改的有序信息流。当您对表启用流时，DynamoDB 将捕获有关对表中的数据项目进行的每项修改的信息。DynamoDB 在 AWS Lambda 上运行，因此您可以创建触发器 - 自动响应 DynamoDB 流中的事件的代码片段。利用触发器，您可以创建应对 DynamoDB 表中的数据修改的应用程序。

当您对表启用流时，您可以将流 Amazon Resource Name (ARN) 与您编写的 Lambda 函数关联起来。在修改表中的项目之后，表的流中都将出现一条新记录。AWS Lambda 将轮询流并在检测到新的流记录时同步调用 Lambda 函数。

Amazon SageMaker 和 Amazon SageMaker 服务终端节点

Amazon SageMaker 是一个完全托管式平台，使开发人员和数据科学家能够以任何规模快速构建、训练和部署机器学习 (ML) 模型。SageMaker 包含多个模块，这些模块可用于共同或单独构建、训练以及部署机器学习 (ML) 模型。借助 Amazon SageMaker 服务终端节点，您可以使用在 Amazon SageMaker 内部或外部开发的已部署模型创建托管式终端节点，以进行实时推理。

通过利用 AWS 软件开发工具包，您可以调用 SageMaker 终端节点来传递内容类型信息及内容，然后根据传递的数据接收实时预测。这样，您就能够将机器学习 (ML) 模型的设计和开发与对推理的结果执行操作的代码分开。

这使数据科学家能够专注于机器学习 (ML)，而使用机器学习 (ML) 模型的开发人员可以专注于如何在代码中使用此模型。有关如何在 SageMaker 中调用终端节点的更多信息，请参阅 Amazon SageMaker API 参考中的 InvokeEnpoint。

实时推理数据洞察

前面的架构图显示，Fast Sneakers 的现有 Web 应用程序添加了包含点击流事件的 Kinesis 数据流，该数据流提供来自网站的流量和事件数据。产品目录（包含分类、产品属性和定价等信息）和订单表（包含已订购商品、账单、配送等数据）是单独的 DynamoDB 表。数据流源和相应的 DynamoDB 表在 AWS Glue Data Catalog 中定义了元数据和架构，供 AWS Glue 流式处理 ETL 任务使用。

通过在 AWS Glue 流式处理 ETL 任务中利用 Apache Spark、Spark Streaming 和 `DynamicFrames`，Fast Sneakers 能够从任一数据流中提取数据并进行转换，同时合并来自产品表和订单表的数据。利用来自此转换的水合数据，用于从中获取推理结果的数据集将提交到 DynamoDB 表。

该表的 DynamoDB 流会为写入的每条新记录触发一个 Lambda 函数。Lambda 函数使用 AWS 软件开发工具包将之前转换的记录提交到 SageMaker 终端节点，以推理产品需要进行怎样的价格调整（如果有的话）。如果机器学习 (ML) 模型确定需要对价格进行调整，则 Lambda 函数会将价格更改写入目录 DynamoDB 表中的产品。

总结

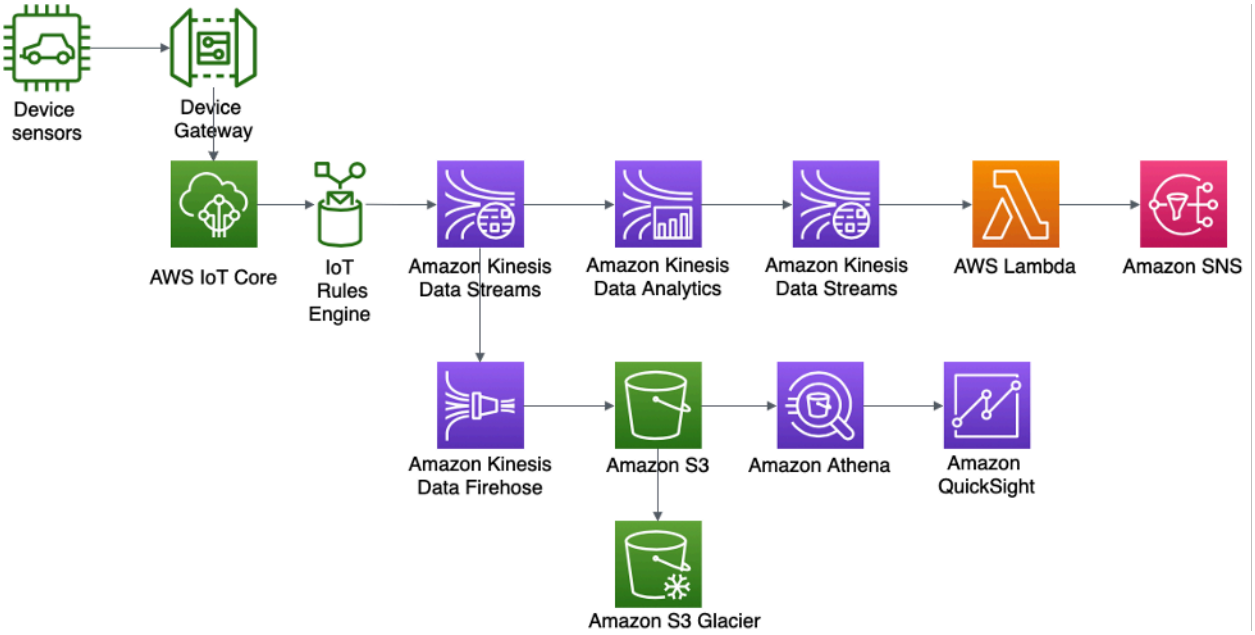
Amazon Kinesis Data Streams 可让您轻松地收集、处理和分析实时流数据，以便您及时获得洞察并对新信息快速做出响应。与 AWS Glue 无服务器数据集成服务相结合，您可以创建实时事件流应用程序，为机器学习 (ML) 准备和合并数据。

由于 Kinesis Data Streams 和 AWS Glue 服务都是完全托管的，因此 AWS 消除了为大数据平台管理基础设施的无差别繁重工作，让您专注于基于数据生成数据洞察。

Fast Sneakers 可以利用实时事件处理和机器学习 (ML) 使他们的网站能够进行完全自动的实时价格调整，从而最大限度地利用产品库存。这为他们的业务带来了最大的价值，同时避免了创建和维护大数据平台的需要。

场景 4：设备传感器实时异常检测和通知

ABC4Logistics 公司将汽油、液态丙烷 (LPG) 和石脑油等高度易燃的石油产品从港口运输到各个城市。数百辆车上安装了多个传感器，用于监控位置、发动机温度、集装箱内的温度、行驶速度、停车位置、路况等情况。ABC4Logistics 的其中一项要求是实时监控发动机和集装箱的温度，并在出现任何异常情况时提示驾驶员和车队监控团队。为了实时检测此类情况并生成提示，ABC4Logistics 在 AWS 上实施了以下架构。



ABC4Logistics 的设备传感器实时异常检测和通知架构

来自设备传感器的数据由 AWS IoT Gateway 摄取，其中，AWS IoT 规则引擎将在 Amazon Kinesis Data Streams 中提供流数据。使用 Kinesis Data Analytics，ABC4Logistics 可以对 Kinesis Data Streams 中的流数据执行实时分析。

使用 Kinesis Data Analytics，ABC4Logistics 可以检测来自传感器的温度读数是否在十秒期间内偏离了正常读数，并将记录摄取到另一个 Kinesis Data Streams 实例中，从而识别异常记录。然后，Amazon Kinesis Data Streams 会调用 Lambda 函数，这些函数可以通过 Amazon SNS 向驾驶员和车队监控团队发送提示。

Kinesis Data Streams 中的数据也会向下推送到 Amazon Kinesis Data Firehose 中。Amazon Kinesis Data Firehose 将这些数据保留在 Amazon S3 中，从而允许 ABC4Logistics 对传感器数据执行批处理分析或近实时分析。ABC4Logistics 使用 Amazon Athena 查询 S3 中的数据，并使用 Amazon QuickSight 进行可视化。为了进行长期数据留存，S3 生命周期策略用于将数据归档到 Amazon S3 Glacier。

接下来将详细介绍此架构的重要组成部分。

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics 使您能够转换和分析流数据，并实时响应异常情况。它是 AWS 上的无服务器服务，这意味着 Kinesis Data Analytics 负责预置，并弹性地扩展基础设施以处理任何数据吞吐量。这就省去了设置和管理流基础设施的所有无差别的繁重工作，使您能够将更多时间花在编写流应用程序上。

借助 Amazon Kinesis Data Analytics，您可以使用多个选项（包括标准 SQL 以及 Java、Python 和 Scala 中的 Apache Flink 应用程序）以交互方式查询流数据，还可以使用 Java 构建 Apache Beam 应用程序来分析数据流。

这些选项使您可以灵活地使用特定方法，具体取决于流式处理应用程序和源/目标支持的复杂程度。以下部分将讨论适用于 Flink 应用程序的 Kinesis Data Analytics 选项。

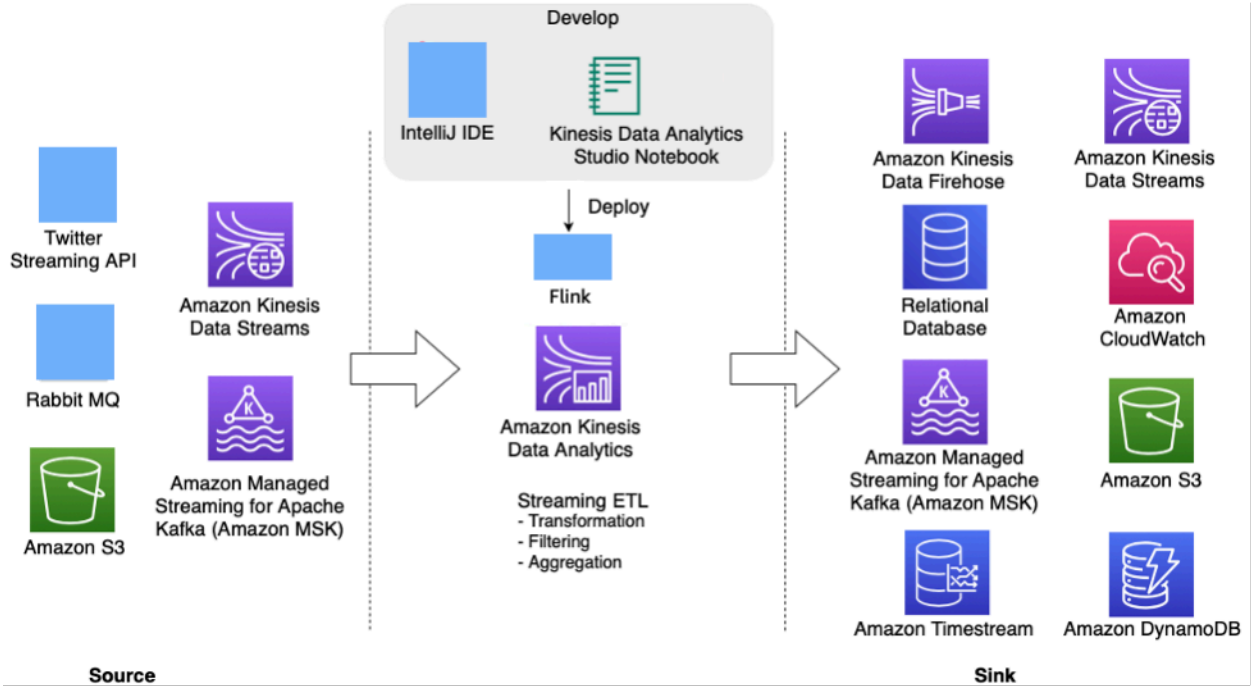
适用于 Apache Flink 应用程序的 Amazon Kinesis Data Analytics

Apache Flink 是一种常用的开源框架和分布式处理引擎，用于对无界和有界数据流进行有状态计算。Apache Flink 旨在以内存中速度大规模执行计算，并支持 exactly-one（精确一个）语义。基于 Apache Flink 的应用程序以容错方式帮助实现低延迟和高吞吐量。

借助 Amazon Kinesis Data Analytics for Apache Flink，您可以在不管理复杂的分布式 Apache Flink 环境的情况下，针对流源编写和运行代码，以执行时间序列分析、向实时控制面板提供数据以及创建实时指标。您可以使用高级 Flink 编程功能，使用方式与自行托管 Flink 基础设施时一样。

借助 Kinesis Data Analytics for Apache Flink，您可以在 Java、Scala、Python 或 SQL 中创建应用程序来处理和分析流数据。典型的 Flink 应用程序从输入流或数据位置或源 读取数据，使用运算符或函数转换/筛选或联接数据，然后将数据存储在输出流或数据位置或接收器 上。

下面的架构图显示了 Kinesis Data Analytics Flink 应用程序支持的一些源和接收器。除了用于源/接收器的预先捆绑的连接器之外，还可以为 Kinesis Data Analytics 上的 Flink 应用程序的各种其他源/接收器引入自定义的连接器。



Kinesis Data Analytics 上用于实时流式处理的 Apache Flink 应用程序

开发人员可以使用他们首选的 IDE 来开发 Flink 应用程序，并通过 AWS Management Console或 DevOps 工具将它们部署在 Kinesis Data Analytics 上。

Amazon Kinesis Data Analytics Studio

作为 Kinesis Data Analytics 服务的一部分，Kinesis Data Analytics Studio 可供客户实时以交互方式查询数据流，并使用 SQL、Python 和 Scala 轻松构建和运行流式处理应用程序。Studio 笔记本由 Apache Zeppelin 提供支持。

使用 Studio 笔记本，您可以在笔记本环境中开发 Flink 应用程序代码，实时查看代码的结果，并在笔记本中将其可视化。只需在 Kinesis Data Streams 和 Amazon MSK 控制台中单击一下，即可创建由 Apache Zeppelin 和 Apache Flink 提供支持的 Studio 笔记本，也可以从 Kinesis Data Analytics 控制台启动它。

将代码作为 Kinesis Data Analytics Studio 的一部分迭代进行开发后，您可以将笔记本部署为 Kinesis 数据分析应用程序，以便在流模式下持续运行，同时从源读取数据、写入目标位置、维护长时间运行的应用程序状态以及基于源流的吞吐量自动扩缩。早些时候，客户已使用适用于 SQL 应用程序的 Kinesis Data Analytics 对 AWS 上的实时流式处理数据进行此类交互式分析。

适用于 SQL 应用程序的 Kinesis Data Analytics 仍然可用，但对于新项目，AWS 建议您使用新的 Kinesis Data Analytics Studio。Kinesis Data Analytics Studio 将易用性与高级分析功能相结合，助您在几分钟内即可打造出成熟而完善的流式处理应用程序。

为了使 Kinesis Data Analytics Flink 应用程序具有容错性，您可以利用检查点操作和快照，如在 Kinesis Data Analytics for Apache Flink 中实施容错能力中所述。

Kinesis Data Analytics Flink 应用程序对于编写复杂的流式分析应用程序（例如数据处理精确一次 (exactly-once) 语义的应用程序）、执行检查点操作以及处理来自 Kinesis Data Streams、Kinesis Data Firehose、Amazon MSK、Rabbit MQ 和 Apache Cassandra（包括自定义连接器）等数据源的数据非常有用。

在 Flink 应用程序中处理流数据后，您可以将数据保存到各种接收器或目标位置，例如 Amazon Kinesis Data Streams、Amazon Kinesis Data Firehose、Amazon DynamoDB、Amazon OpenSearch Service、Amazon Timestream、Amazon S3 等。此外，Kinesis Data Analytics Flink 应用程序还提供亚秒级性能保证。

适用于 Kinesis Data Analytics 的 Apache Beam 应用程序

Apache Beam 是一种用于处理流数据的编程模型。Apache Beam 提供了一个可移植的 API 层，用于构建成熟完善的数据并行处理管道，这些管道可以在各种引擎或运行器（如 Flink、Spark Streaming、Apache Samza 等）上运行。

您可以将 Apache Beam 框架与 Kinesis 数据分析应用程序结合使用，以处理流数据。使用 Apache Beam 的 Kinesis 数据分析应用程序使用 Apache Flink 运行器来运行 Beam 管道。

总结

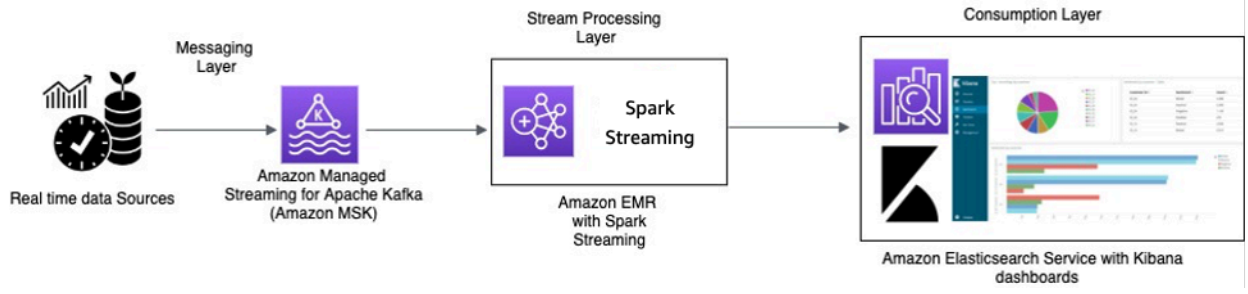
通过利用 AWS 流式处理服务 Amazon Kinesis Data Streams 、Amazon Kinesis Data Analytics 和 Amazon Kinesis Data Firehose，

ABC4Logistics 可以检测温度读数中的异常模式，并实时通知驾驶员和车队管理团队，防止发生如整车故障或火灾等重大事故。

场景 5：使用 Apache Kafka 进行实时遥测数据监控

ABC1Cabs 是一家在线出租车预订服务公司。所有出租车都有 IoT 设备，可从车辆收集遥测数据。目前，ABC1Cabs 正在运行 Apache Kafka 集群，这些集群用于记录实时事件使用情况、收集系统运行状况指标、活动跟踪，以及将数据提供给在 Hadoop 集群上内部构建的 Apache Spark Streaming 平台。

ABC1Cabs 使用 OpenSearch Dashboards 显示业务指标、进行调试、发出提示和创建其他控制面板。他们对 Amazon MSK、带有 Spark Streaming 的 Amazon EMR 和带有 OpenSearch Dashboards 的 OpenSearch Service 感兴趣。他们的要求是减少维护 Apache Kafka 和 Hadoop 集群的管理开销，同时使用熟悉的开源软件和 API 来编排其数据管道。以下架构图显示了他们在 AWS 上的解决方案。



使用 Amazon MSK 进行实时处理，并使用 Amazon EMR 上的 Apache Spark Streaming 和带有 OpenSearch Dashboards 的 Amazon OpenSearch Service 进行流式处理

出租车 IoT 设备收集遥测数据并发送到源中心。源中心配置为实时向 Amazon MSK 发送数据。使用 Apache Kafka 生成者库 API，Amazon MSK 配置为将数据流式传输到 Amazon EMR 集群中。Amazon EMR 集群安装了 Kafka 客户端和 Spark Streaming，以便能够使用和处理数据流。

Spark Streaming 具有接收器连接器，它们可以将数据直接写入 Elasticsearch 的已定义索引。带有 OpenSearch Dashboards 的 Elasticsearch 集群可用于指标和控制面板。Amazon MSK、带有 Spark Streaming 的 Amazon EMR 以及带有 OpenSearch Dashboards 的 OpenSearch Service 都是托管式服务，在这些服务中，AWS 负责管理不同集群的基础设施管理方面的无差别繁重工作，这使您只需单击几下即可使用熟悉的开源软件构建应用程序。下一节将详细介绍这些服务。

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Apache Kafka 是一个开源平台，使客户能够捕获流数据，例如点击流事件、交易、IoT 事件以及应用程序和机器日志。利用这些信息，您可以开发用于执行实时分析、运行持续转换以及将这些数据实时分发到数据湖和数据库的应用程序。

您可以使用 Kafka 作为流数据存储以将应用程序与生成者和使用者分离，并在两个组件之间实现可靠的数据传输。尽管 Kafka 是一种通用的企业级数据流式处理和消息收发平台，但在生产环境中设置、扩展和管理它可能很困难。

Amazon MSK 负责这些管理任务，并使您能够在遵循高可用性和安全性最佳实践的环境中轻松设置、配置和运行 Kafka 以及 Apache Zookeeper。您仍然可以使用 Kafka 的控制层面操作和数据层面操作来管理数据的生成和使用。

由于 Amazon MSK 运行和管理开源 Apache Kafka，因此客户可以轻松地在 AWS 上迁移和运行现有 Apache Kafka 应用程序，而无需对其应用程序代码进行更改。

扩缩

Amazon MSK 提供扩缩操作，以便用户可以在集群运行时主动扩展集群。创建 Amazon MSK 集群时，您可以在集群启动时指定代理的实例类型。您可以从 Amazon MSK 集群中的若干代理开始。然后，您可以使用 AWS Management Console或 AWS CLI 纵向扩展到每个集群数百个代理。

或者，您可以通过更改 Apache Kafka 代理的大小或系列来扩展集群。更改代理的大小或系列让您能够灵活地调整 Amazon MSK 集群的计算容量，来应对工作负载的变化。使用 Amazon MSK 大小和定价电子表格（文件下载）以确定适用于您的 Amazon MSK 集群的正确代理数量。此电子表格提供与类似的、自行管理的基于 EC2 的 Apache Kafka 集群相比，估计的 Amazon MSK 集群大小和相关 Amazon MSK 成本。

创建 Amazon MSK 集群后，您可以增加每个代理的 EBS 存储量，但减少存储除外。在此纵向扩展操作期间，存储卷仍然可用。它提供两种类型的扩缩操作：弹性伸缩和手动扩缩。

Amazon MSK 支持使用 Application Auto Scaling 策略自动扩展集群的存储，以响应使用量的增加。您的弹性伸缩策略会设置目标磁盘利用率和最大扩缩容量。

存储利用率阈值可帮助 Amazon MSK 触发弹性伸缩操作。要使用手动扩缩来增加存储空间，请等待集群进入 ACTIVE 状态。在两次事件之间，存储扩缩的冷却时间至少为六小时。尽管该操作可立即提供更多存储，但该服务仍会对集群执行优化，这可能需要长达 24 小时或更长时间。

这些优化的持续时间与您的存储大小成正比。此外，它还在 AWS 区域内提供多可用区复制功能，以提供高可用性。

配置

Amazon MSK 提供代理、主题和 Apache ZooKeeper 节点的原定设置配置。您还可以创建自定义配置，并使用这些配置来创建新的 Amazon MSK 集群或更新现有集群。当您在未指定自定义 Amazon MSK 配置的情况下创建 MSK 集群时，Amazon MSK 会创建并使用原定设置配置。有关原定设置值的列表，请参阅 Apache Kafka 配置。

出于监控目的，Amazon MSK 会收集 Apache Kafka 指标并将其发送到 Amazon CloudWatch，您可以在其中查看它们。系统会自动收集您为 MSK 集群配置的指标并将其推送给 CloudWatch。通过监控使用者延迟，您可以确定那些未能及时获得某个主题中可用的最新数据的滞后或停滞的使用者。必要时，您可以采取补救措施，例如扩缩或重新启动这些使用者。

迁移到 Amazon MSK

从本地部署迁移到 Amazon MSK 可以通过以下方法之一实现。

- MirrorMaker2.0 - MirrorMaker2.0 (MM2) MM2 是一款基于 Apache Kafka Connect 框架的多集群数据复制引擎。MM2 是 Apache Kafka 源连接器和接收器连接器的组合。您可以使用单个 MM2 集群在多个集群之间迁移数据。MM2 自动检测新的主题和分区，同时还确保主题配置在集群之间同步。MM2 支持迁移 ACL、主题配置和偏移转换。有关迁移的更多详细信息，请参阅使用 Apache Kafka 的 MirrorMaker 迁移集群。MM2 用于与自动复制主题配置和偏移转换相关的使用案例。
- Apache Flink - MM2 支持至少一次 (at least once) 语义。记录可以复制到目标位置，并且使用者在处理重复的记录时应该是幂等的。在精确一次 (exactly-once) 场景中，语义是必需的，客户可以使用 Apache Flink。它提供了一种实现精确一次 (exactly-once) 语义的替代方法。

  Apache Flink 还可用于数据在提交到目标集群之前需要进行映射或转换操作的场景。Apache Flink 为 Apache Kafka 提供了带有源和接收器的连接器，这些源和接收器可以从一个 Apache Kafka 集群读取数据并写入另一个集群。Apache Flink 可以通过以下方式在 AWS 上运行：启动 Amazon EMR 集群，或使用 Amazon Kinesis Data Analytics 将 Apache Flink 作为应用程序运行。
- AWS Lambda - 由于支持将 Apache Kafka 作为 AWS Lambda 的事件源，客户现在可以通过 Lambda 函数使用主题中的消息。AWS Lambda 服务在内部轮询来自事件源的新记录或消息，然后

同步调用目标 Lambda 函数来使用这些消息。Lambda 批量读取消息，并在事件有效负载中向函数提供消息批次以供处理。然后，所使用的消息可以转换和/或直接写入您的目标 Amazon MSK 集群。

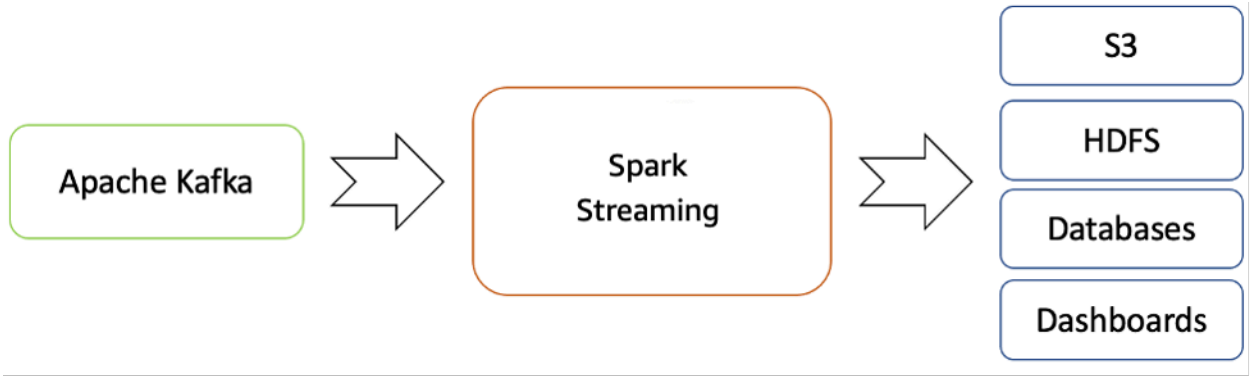
具有 Spark Streaming 的 Amazon EMR

Amazon EMR 是一个托管式集群平台，可简化在 AWS 上运行大数据框架（如 Apache Hadoop 和 Apache Spark）以处理和分析海量数据的操作。

Amazon EMR 提供了 Spark 的功能，可用于启动 Spark Streaming 以使用来自 Kafka 的数据。Spark Streaming 是核心 Spark API 的扩展，支持对实时数据流进行可扩展、高吞吐量、容错的流式处理。

您可以使用 AWS Command Line Interface (AWS CLI) 或在 AWS Management Console上创建 Amazon EMR 集群，并在创建此集群时在高级配置中选择 Spark 和 Zeppelin。如下面的架构图所示，可以从许多来源（如 Apache Kafka 和 Kinesis Data Streams）提取数据，并且可以使用由高级函数（如 map、reduce、join 和 window）表示的复杂算法进行处理。有关更多信息，请参阅 DStreams 上的转换。

处理后的数据可以向外推送到文件系统、数据库和实时控制面板。



从 Apache Kafka 到 Hadoop 生态系统的实时流式处理流

原定设置情况下，Apache Spark Streaming 具有微批处理运行模型。但是，自 Spark 2.3 推出以来，Apache 引入了一种名为连续处理的新的低延迟处理模式，该模式可以在保证 at-least-once（至少一次）语义处理的情况下实现低至一毫秒的端到端延迟。

在不更改查询中的 Dataset/DataFrames 操作的情况下，您可以根据应用程序要求选择此模式。Spark Streaming 的一些益处包括：

- 它将 Apache Spark 的语言集成 API 引入到流式处理中，可让您像编写批处理任务一样编写流任务。
- 它支持 Java、Scala 和 Python。

- 它可以通过开箱即用的方式恢复丢失的工作和操作员状态（如滑动窗口），而无需任何额外的代码。
- 通过在 Spark 上运行，Spark Streaming 可让您重复使用相同的代码进行批处理，根据历史数据联接流，或对流状态运行即席查询，并构建功能强大的交互式应用程序，而不仅仅是分析。
- 使用 Spark Streaming 处理数据流后，可以使用 OpenSearch 接收器连接器将数据写入 OpenSearch Service 集群，反过来，可以将带有 OpenSearch Dashboards 的 OpenSearch Service 用作使用层。

带有 OpenSearch Dashboards 的 Amazon OpenSearch Service

OpenSearch Service 是一种托管式服务，可以让您轻松地在 AWS 云中部署、操作和扩展 OpenSearch 集群。OpenSearch 是一款通用的开源搜索和分析引擎，适用于日志分析、实时应用程序监控、点击流分析等使用案例。

OpenSearch Dashboards 是一种开源数据可视化和挖掘工具，可以用于日志和时间序列分析、应用程序监控和运营智能使用案例。它提供了强大且易用的功能，例如直方图、线形图、饼图、热图和内置的地理空间支持。

OpenSearch Dashboards 提供了与 OpenSearch（一款常用的分析和搜索引擎）的紧密集成，这使 OpenSearch Dashboards 成为用于可视化存储在 OpenSearch 中的数据的原定设置选择。OpenSearch Service 为每个 OpenSearch Service 域提供 OpenSearch Dashboards 安装。您可以在 OpenSearch Service 控制台的域控制面板上找到指向 OpenSearch Dashboards 的链接。

总结

使用在 AWS 上作为托管式服务提供的 Apache Kafka，您可以将重点放在使用方面，而不是管理代理之间的协调（这通常需要详细了解 Apache Kafka）。高可用性、代理可扩展性和精细访问控制等功能由 Amazon MSK 平台进行管理。

ABC1Cabs 利用了这些服务来构建生产应用程序，而无需具备基础设施管理专业知识。他们可以专注于处理层来使用 Amazon MSK 中的数据并进一步传播到可视化层。

Amazon EMR 上的 Spark Streaming 可以帮助实时分析流数据，并在 Amazon OpenSearch Service 中的 OpenSearch Dashboards 上发布以供可视化层使用。

结论和贡献者

结论

本文档回顾了流式处理工作流的几种场景。在这些场景中，流数据处理为示例公司提供了添加新特性和功能的能力。

通过在创建数据时对数据进行分析，您将深入了解您的企业当前在做什么。借助 AWS 流服务，您可以专注于应用程序以制定出对时间敏感的业务决策，而不是部署和管理基础设施

贡献者

- Amalia Rabinovitch，AWS 高级解决方案构架师
- Priyanka Chaudhary，AWS 数据湖数据构架师
- Zohair Nasimi，AWS 解决方案构架师
- Rob Kuhr，AWS 解决方案构架师
- Ejaz Sayyed，AWS 合作伙伴解决方案高级构架师
- Allan MacInnis，AWS 解决方案构架师
- Chander Matrubhutam，AWS 产品营销经理

文档修订

要获得有关此白皮书更新的通知，请订阅 RSS 源。

更新-历史记录-更改	更新-历史记录-描述	更新-历史记录-日期
已更新	更新了技术准确性相关内容	2021 年 9 月 1 日
初次发布	白皮书首次发布	2017 年 7 月 1 日