Kafka e o StreamSets

Kafka, o que é?

Kafka é um sistema de mensagens distribuído. A página da Apache (<https://kafka.apache.org/intro>) relacionada ao projeto define o Kafka como um sistema equivalente ao sistema nervoso central do ser humano, é a base tecnológica para o Always-on. Falando tecnicamente, Kafka é só o termo comummente conhecido para Event Stream, ou Transmissão de Eventos, que nada mais é do que a prática de capturar eventos no momento em que eles acontecem (real-time) de sistemas como bancos de dados, sensores, telefones celulares, serviços de nuvem e outros softwares que se utilizam dessa característica de envio de eventos. Enquanto no Kafka, esses eventos são armazenados em tópicos por um período determinado e, após a passagem desse período, essa informação é apagada. O Streaming de Eventos é utilizado para muitos fins como comentei anteriormente, e esse tipo de mensageria traz um dinamismo ao nosso dia a dia, com ela é possível acompanhar uma transação bancária e identificar uma fraude, mapear um comportamento anômalo em um sensor ou sistema, entre outros usos.

StreamSets, para que serve?

StreamSets é uma ferramenta cloud-native utilizada para construir, executar e monitorar pipelines de dados.

Um pipeline descreve o fluxo do dado desde a sua origem até os destinos e define como o dado será processado ao longo do caminho. Esses pipelines podem acessar múltiplos tipos de sistemas, incluindo Data Lakes, Data Warehouse, sistemas de armazenamento instalados em ambientes on-premises como bancos de dados relacionais. Conforme o pipeline é executado, é possível visualizar estatísticas em tempo real, informações sobre erros em dados desde a sua origem até o destino. Basicamente, o StreamSets é uma ferramenta de transformação de dados, sejam eles gerados à partir de tópicos Kafka (Streaming de Eventos) como no nosso exemplo à seguir, bancos de dados relacionais para extração de dados utilizando ETL/ELT em bulk/batch, e até mesmo Change Data Capture (CDC).

Bom, agora que já passamos pelo overview da solução, vamos ao passo a passo do meu lab.

Pre-requisitos:

Antes de começar a mexer no StreamSets e Kafka, precisaremos de 3 aplicações que podem ser instanciadas em IBM Cloud e utilizadas de forma gratuita, são o DB2, EventStream e Cloud Object Storage ambos em Dallas.

EventStreams

No EventStreams após instanciar a solução, faça a criação de uma credencial de serviço, copie todo o conteúdo da credencial e salve-a em uma pasta local no seu computador com o nome **ibm\_credentials.json**, guarde o local do arquivo porque precisaremos dele no script Python.

A screenshot of a computer

AI-generated content may be incorrect.

COS (Cloud Object Storage)

Ao concluir o provisionamento do COS, crie uma instancia onde armazenaremos os dados e crie também uma credencial de serviço. Na home da sua Instancia, procure por Service Credentials

A screenshot of a computer

AI-generated content may be incorrect.

Do json que a credencial gera, precisaremos dos campos access\_key\_id e secret\_access\_key.

Acesse a sua instância e, dentro do bucket, procure pela opção Configuration. Dessa página precisaremos do Public Endpoint, se ele não estiver aparecendo, role a página para baixo.

DB2 - Opcional:

Se quiser usar o DB2 para armazenar os dados, alguns passos a mais serão necessários. Aqui, na IBM Cloud, armazene os dados gerados na credencial de serviço em um notepad. Ao final do lab, inclui uma opção para utilizar o DB2.

A screenshot of a computer

AI-generated content may be incorrect.

Vamos começar a validar se o java está instalado, se não, recomendo instalar a versão 17.

Java – version:A screenshot of a computer

AI-generated content may be incorrect.

Na primeira parte desse lab, iremos configurar o Kafka. Como estou usando Mac, utilizei o brew para fazer a instalação do Kafka e start dos serviços, siga o passo a passo a seguir. Se você não estiver usando o brew para instalar o Kafka, siga adiante até a seção de Kafka no Terminal.

1 - Para instalar o Kafka:

brew install kafka

2 - Após a instalação, para inicialização do Kafka, utilize o comando:

brew services start kafka

3 - Se quiser validar se o serviço está rodando, no Mac o comando é sua tela deve mostrar a mensagem como a seguir:

brew services info –all

A screenshot of a computer

AI-generated content may be incorrect.

4 - Kafka funcionando! Agora vamos criar o tópico:

kafka-topics --create --topic process\_events --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

5 - Pronto, o tópico Kafka está aguardando as informações.

Se você quiser ver ele funcionando, e estiver testando localmente, utilize esse comando:

kafka-console-consumer --bootstrap-server localhost:9092 --topic process\_events

Sua tela deve ficar com o cursor piscando, como na imagem à seguir:

A computer screen with white text

AI-generated content may be incorrect.

Se você estiver usando o Kafka diretamente no terminal, siga esse passo a passo. Baixe o Kafka da web e descompacte em uma pasta do seu computador, abra o terminal e navegue até essa pasta. Se o apache foi baixado do site da Apache, execute o seguinte comando. O nome do seu arquivo pode ser diferente do meu:

tar -xvf kafka-3.9.1-src.tgz

A screenshot of a computer

AI-generated content may be incorrect.

Navegue até a pasta bin do Kafka

A screenshot of a computer

AI-generated content may be incorrect.

Dessa forma antes de iniciarmos o Kafka, precisamos subir o zookeeper, para isso execute o seguinte comando:

#iniciar zookeeper

./zookeeper-server-start.sh ../config/zookeeper.properties

A screenshot of a computer screen

AI-generated content may be incorrect.

Abra um novo terminal e vamos iniciar o Kafka

#iniciar broker

./kafka-server-start.sh ../config/server.properties

A screen shot of a computer screen

AI-generated content may be incorrect.

Volte ao passo 4 da instrução do Brew para continuar.

Agora, copie o arquivo log\_generator.py, e o coloque junto ao json de credenciais e o eventlog de base:

A screenshot of a computer program

AI-generated content may be incorrect.

Abra um novo terminal, e execute o comando

python3 log\_genrator.py

Sua tela será exibida da seguinte forma, e se as credenciais estiverem certas, o Kafka já emitirá eventos.

A screen shot of a computer

AI-generated content may be incorrect.

Se quiser testar se o tópico está emitindo eventos, execute o seguinte comando:

kafka-console-consumer --bootstrap-server localhost:9092 --topic process\_events

StreamSets

Não iremos criar um Environment porque estou inferindo que, ao solicitar o acesso a uma conta POC, o Environment é automaticamente criado. Se o seu ambiente não possuir um environment, siga o passo a passo a seguir para criar, se você já for atribuído a um, siga para Clonando o Ambiente à seguir:

Clonando o Ambiente

No deploy disponível, procure pelo kebab (as três bolinhas no canto direito da tela) e clique em clonar, recomendo colocar as suas iniciais no nome do deployment, no meu caso TNM e as tags que ajudarão a filtrar o deploy posteriormente. Selecione a versão da engine, recomendo a 6.2.1, mas pode ser qualquer outra acima dela.

A screenshot of a computer

AI-generated content may be incorrect.

Como você realizou o clone de um environment que já existe, recomendo antes de iniciar o Deploy, adicionar as libs necessárias para o nosso laboratório. Na tela de deployment, selecione o kebab e clique em Editar. Role a tela para baixo até visualizar a opção Configure Engine e clique na opção Stage Libraries.

A screenshot of a computer

AI-generated content may be incorrect.

Na tela que será exibida, selecione as libs clicando no botão de + azul ao lado de cada uma delas, ou do lado esquerdo em Available Stages clique no botão Select > All Stages. Selecione o quadrado na linha de cabeçalhos, ao lado da opção Stage Library e clique Ok. Clique Ok e Save. Volte ao Kebab e clique em Start.

A screenshot of a computer

AI-generated content may be incorrect.

Nesse laboratório não irei abordar o deployment de uma engine, infiro que já tenha realizado os cursos oficiais e saiba realizar o deployment de uma remote engine.

Criando as conexões

Nesse primeiro momento, não vamos adicionar uma engine a conexão, nomeie a sua conexão, preencha as informações conforme à seguir:

* Type: Kafka
* brokerURL: Insira a lista de brokers nesse campo sem as “” e separados por vírgula (“,”)
* Security (Coloque as informações conforme o seu arquivo bootstrap.ini
  + Security Option: Custom Authentication (Security Protocol = Custom)
* Adicione 6 Propriedades custom e copie as informações do seu arquivo de configurações baixado do EventStream:
  + sasl.jaas.config
  + security.protocol
  + sasl.mechanism
  + ssl.protocol
  + ssl.enabled.protocols
  + ssl.endpoint.identification.algorithm

Ao final, clique eu TestConnection (Apache Kafka)

A screenshot of a computer

AI-generated content may be incorrect.

Após criar as conexões, navegue até o menu Build do lado esquerdo da tela e clique em Pipelines. Crie um novo pipeline, para atender a completude do nosso lab, vamos inserir os seguintes estágios na tela, como na imagem abaixo:

* Kafka Multitopic Consumer
* Expression Evaluator
* Field Order
* Amazon S3

A screenshot of a computer

AI-generated content may be incorrect.

Configurando os estagios:

1 - Kafka Multitopic, vamos configurar o estágio da seguinte forma:

* Connection
  + Connection: Selecionar a conexão Kafka que criamos
  + Consumer Group: streamsetsDataCollector
  + Topic Subscription Type: Topics List
  + Topic List: process\_events (valide se esse é o tópico que está no script Python)
* Data Format
  + Data Format: JSON

As demais opções deixem como está.

2 – Expression Evaluator

* Expressions:
  + Output Field:
    - /"Case ID"
    - /Activity
    - /"Event Time"
    - /"Product ID"
  + Field Expression:
    - ${record:value("/Case ID")}
    - ${record:value('/Activity')}
    - ${record:value("/Event Time")}
    - ${record:value("/Product ID")}

3 – Field Order

* Fields to Order (Insira os campos), eles precisam ser inseridos com a barra antes dos campos, se não quiser inserir manualmente, execute um preview e clique nos campos correspondentes pela lupa do lado direito:
  + /"Case ID"
  + /Activity
  + /"Event Time"
  + /"Product ID"

4 – Amazon S3

* AWS S3
  + Connection: Selecione a conexão.
  + Bucket: informe o nome do bucket onde os arquivos serão salvos
  + Object Name Suffix: Insira apenas CSV, sem aspas ou pontos
* Data Format
  + Data Format: Delimited
  + Header line: With Header Line

Bom, até aqui se você estiver com o Kafka em execução desde o começo do nosso lab, provavelmente você terá uma serie de eventos para serem consumidos, ao testar o pipeline do StreamSets você verá o consumo massivo desses dados.

Vamos consumir os dados e vê-los passando pelo pipeline, simulando um gerador de eventos. Clique em Draft Run > Reset Origin & Start, se a configuração estiver correta, depois de alguns segundos você visualizará a seguinte tela:

A screenshot of a computer

AI-generated content may be incorrect.

Se você chegou até aqui, parabéns! Você acaba de concluir o laboratório de Kafka com StreamSets. Agora, tente fazer a conexão com o DB2 que pedi para criar no começo do lab, simule o join do dado que está em trânsito com uma tabela de dados.

A melhor maneira de aprender é colocando em prática. Vou te deixar uma dica de como configurar o DB2 aqui:

* JDBC
  + Connection: none
  + JDBC Connection String: jdbc:db2://<hostname>:<port>/BLUDB
  + SQL Query (Example): SELECT ID\_PRODUCT, DESCRIPTION
  + FROM TZX44874.DB2\_product\_lookup
  + WHERE ID\_PRODUCT = '${record:value("/Product ID")}'
  + Multiple Values Behavior: First value Only
  + Additional JDBC Configuration Properties
    - Name: sslConnection
    - Value: true
  + Deixe o resto como está
  + Adicione o usuário e senha à aba Credentials
  + Baixe o driver JDBC para o DB2 do site <https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-downloads>
  + Descompacte os arquivos, se dentro da pasta extraída houver outro arquivo .zip, descompacte-o com o comando unzip:A screenshot of a computer

    AI-generated content may be incorrect.
  + Pegue o arquivo db2jcc4.jar e carregue-o em external libraries no estágio JDBC Lookup (se estiver usando esse):

A screenshot of a computer

AI-generated content may be incorrect.

Execute o pipeline e veja a sua consulta a base de dados ser utilizada enquanto os dados vindos do Kafka passam pelo pipeline. Uma forma de verificar se a execução está sendo realizada com sucesso é utilizar a opção Preview

A screenshot of a phone

AI-generated content may be incorrect.

Selecione a opção Configure Preview e selecione os checkboxes conforme abaixo, clique em Run Preview

A screenshot of a computer

AI-generated content may be incorrect.

Clicando em cada estágio, você consegue visualizar os dados passando, abaixo o exemplo do dado sendo enriquecido durante o pipeline, quando clico sobre o JDBC Lookup:

A screenshot of a computer

AI-generated content may be incorrect.

Troubleshooting

Se ao adicionar o estágio de DB2 a sua engine apresentar erro. Faça os seguintes passos:

1 – Acesse a máquina onde a sua engine foi instalada

2 – Navegue até o seguinte local

/home/<user>/sdc/streamsets-datacollector-6.2.1/etc

3 – Abra o arquivo

sdc.properties

Procure pela opção flight service e altere a opção

De 🡪 flight.enable = auto

Para 🡪 flight.enable = true

A screenshot of a computer program

AI-generated content may be incorrect.

4 – Reinicie a Engine.

[English]

Kafka and StreamSets

Kafka, what is it?

Kafka is a distributed messaging system. The Apache page (https://kafka.apache.org/intro) that describes the project defines Kafka as equivalent to the human central nervous system, the technological foundation for always-on systems.

Technically speaking, Kafka is the common term used for Event Stream, which is simply the practice of capturing events in real time, as they happen, from systems such as databases, sensors, mobile phones, cloud services, and other software that uses this event-driven approach.

In Kafka, these events are stored in topics for a defined period, and once that period passes, the information is deleted.

Event streaming can be used for many purposes, such as monitoring a banking transaction to identify fraud, mapping abnormal behavior in a sensor or system, among other use cases.

StreamSets, what is it for?

StreamSets is a cloud-native tool used to build, run, and monitor data pipelines.

A pipeline describes the flow of data from its source to its destinations and defines how the data will be processed along the way. These pipelines can connect to multiple types of systems, including Data Lakes, Data Warehouses, and on-premises storage systems like relational databases.

As the pipeline runs, it is possible to view real-time statistics as well as error information from source to target.

StreamSets is essentially a data transformation tool, whether the data comes from Kafka topics (event streaming, as in our example), relational databases using ETL/ELT in bulk/batch, or even Change Data Capture (CDC).

Now that we’ve reviewed the basics of the solution, let’s move on to the step-by-step lab.

Pre-reqs:

Before using StreamSets and Kafka, we will need 3 applications, which can be instantiated on IBM Cloud and used for free: DB2, EventStreams, and Cloud Object Storage (all deployed in Dallas).

EventStreams

In EventStreams, once the solution is instantiated, create a service credential. Copy all its content and save it locally on your computer as ibm\_credentials.json.

Keep track of the file path, as we will need it in the Python script.

A screenshot of a computer

AI-generated content may be incorrect.

COS (Cloud Object Storage)

Once you finish provisioning COS, create an instance for data storage and also generate a service credential.

On the instance home page, look for Service Credentials.

A screenshot of a computer

AI-generated content may be incorrect.

From the credential JSON, we will need the fields access\_key\_id and secret\_access\_key.

In your bucket, open the Configuration tab and copy the Public Endpoint. If it’s not visible, scroll further down the page.

DB2 – Optional:

If you want to use DB2 for storing the data, additional steps will be needed. In IBM Cloud, save the generated credential data into a notepad. At the end of this lab, I’m suggesting an option to use DB2.

A screenshot of a computer

AI-generated content may be incorrect.

Let’s start by checking if Java is installed. If not, I recommend installing version 17.

Java – version:

A screenshot of a computer

AI-generated content may be incorrect.

In the first part of this lab, we will configure Kafka. As I am using Mac, I used Homebrew to install Kafka and start the services. Follow the steps below. If you are not using Homebrew to install Kafka, skip ahead to the Kafka in Terminal section.

1 - To install Kafka:

brew install kafka

2 - After installation, to start Kafka, use the command:

brew services start kafka

3 - To validate if the service is running on Mac, your screen should display the following message:

brew services info –all

A screenshot of a computer

AI-generated content may be incorrect.

4 - Kafka running! Now let’s create the topic:

kafka-topics --create --topic process\_events --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

5 - Done, the Kafka topic is awaiting information.  
If you want to see it working and you are testing locally run the command below. Your screen should show a blinking cursor, as in the following image:

kafka-console-consumer --bootstrap-server localhost:9092 --topic process\_events

A computer screen with white text

AI-generated content may be incorrect.

If you are using Kafka directly in the terminal, follow these steps. Download Kafka from the web and extract it to a folder on your computer. Open the terminal and navigate to this folder. If Apache was downloaded from the Apache website, execute the following command. Your file name may be different from mine:

tar -xvf kafka-3.9.1-src.tgz

Navigate to the Kafka bin folder.

cd ~/workspace/Apache/kafka/bin

Before starting Kafka, we need to start Zookeeper. To do this, run the following command:

./zookeeper-server-start.sh ../config/zookeeper.properties

A screenshot of a computer screen

AI-generated content may be incorrect.

Open a new terminal and start Kafka:

./kafka-server-start.sh ../config/server.properties

A screen shot of a computer screen

AI-generated content may be incorrect.

Go back to step 4 of the Homebrew instructions to continue.

Now, copy the log\_generator.py file and place it in the same directory as the JSON credentials and the base event log.

A screenshot of a computer program

AI-generated content may be incorrect.

Open a new terminal and run:

python3 log\_generator.py

Your screen will look like this, and if the credentials are correct, Kafka will already be emitting events.

A screen shot of a computer

AI-generated content may be incorrect.

StreamSets

We will not create an Environment, because I am assuming that when requesting access to a POC account, the Environment is automatically created. If your environment does not have one, follow the steps below to create it. If you already have one assigned, move on to Cloning the Environment.

Cloning the Environment

In the available deployment, look for the kebab menu (the three dots on the right side of the screen) and click Clone. I recommend adding your initials to the deployment name, in my case TNM, and tags that will help you filter the deployment later. Select the engine version, I recommend 6.2.1, but it can be any version above that.

A screenshot of a computer

AI-generated content may be incorrect.

As you cloned an existing environment, I recommend adding the necessary libraries for our lab before starting the Deployment. On the deployment screen, select the kebab menu and click Edit. Scroll down until you find the Configure Engine option and click Stage Libraries.

A screenshot of a computer

AI-generated content may be incorrect.

In the screen that appears, add the libraries by clicking the blue + button next to each one, or in the left menu under Available Stages, click Select > All Stages. Select the checkbox next to Stage Library and click Ok. Click Ok and Save. Return to the kebab menu and click Start.

A screenshot of a computer

AI-generated content may be incorrect.

OBS: In this lab, I will not cover the deployment of an engine, assuming you have already taken the official courses and know how to deploy a remote engine.

**Creating the Connections**

At this first moment, we will not add an engine to the connection. Name your connection and fill in the information as follows:

* Type: Kafka
* brokerURL: enter the list of brokers in this field without "" and separated by commas (,)
* Security (information according to your bootstrap.ini file)
  + Security Option: Custom Authentication (Security Protocol = Custom)
* Add 6 custom properties from the EventStream configuration file you downloaded:
  + sasl.jaas.config
  + security.protocol
  + sasl.mechanism
  + ssl.protocol
  + ssl.enabled.protocols
  + ssl.endpoint.identification.algorithm

At the end, click TestConnection (Apache Kafka)

A screenshot of a computer

AI-generated content may be incorrect.

**Building the Pipeline**

After creating the connections, navigate to the Build menu on the left and click Pipelines. Create a new pipeline. To complete our lab, insert the following stages on the canvas:

* Kafka Multitopic Consumer
* Expression Evaluator
* Field Order
* Amazon S3

A screenshot of a computer

AI-generated content may be incorrect.

**Configuring the Stages**

1 - Kafka Multitopic

* Connection: select the Kafka connection we created
* Consumer Group: streamsetsDataCollector
* Topic Subscription Type: Topics List
* Topic List: process\_events (validate if this is the topic in the Python script)
* Data Format: JSON  
  (Leave the other options as default)

2 - Expression Evaluator

* Expressions:
  + Output Field:
    - /"Case ID"
    - /Activity
    - /"Event Time"
    - /"Product ID"
  + Field Expression:
    - ${record:value("/Case ID")}
    - ${record:value('/Activity')}
    - ${record:value("/Event Time")}
    - ${record:value("/Product ID")}

**3 - Field Order**

* Fields to Order (insert fields, always starting with /):
  + /"Case ID"
  + /Activity
  + /"Event Time"
  + /"Product ID"

**4 - Amazon S3**

* Connection: select the connection
* Bucket: enter the name of the bucket where the files will be saved
* Object Name Suffix: enter only CSV, without quotes or dots
* Data Format: Delimited
* Header Line: With Header Line

Up to this point, if Kafka has been running since the start of our lab, you likely already have a series of events to be consumed. When testing the StreamSets pipeline, you will see this massive data consumption.

Click Draft Run > Reset Origin & Start, and if the configuration is correct, after a few seconds you will see the following screen.

A screenshot of a computer

AI-generated content may be incorrect.

If you made it this far, congratulations! You have just completed the Kafka with StreamSets lab.

Now try connecting with the DB2 instance created at the beginning of the lab and simulate a join between the streaming data and a reference table.

**Configuring DB2 (Example)**

* JDBC
* Connection: none
* JDBC Connection String: jdbc:db2://<hostname>:<port>/BLUDB
* SQL Query (Example): SELECT ID\_PRODUCT, DESCRIPTION FROM TZX44874.DB2\_product\_lookup WHERE ID\_PRODUCT = '${record:value("/Product ID")}'
* Multiple Values Behavior: First value only
* Additional JDBC Configuration Properties:
  + Name: sslConnection
  + Value: true

Add username and password in the Credentials tab.

Download the JDBC driver for DB2 from: <https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-downloads>

Extract the files. If an additional .zip file is found inside, extract it with:

unzip <file>.zip

A screenshot of a computer

AI-generated content may be incorrect.

Take the file db2jcc4.jar and upload it into external libraries of the JDBC Lookup stage.

A screenshot of a computer

AI-generated content may be incorrect.

Run the pipeline and observe your query being used while Kafka data flows through the pipeline. You can check success with the Preview option.

A screenshot of a phone

AI-generated content may be incorrect.

Check the options on Configure like the image below:

A screenshot of a computer

AI-generated content may be incorrect.

Clicking on each stage provides you an overview of the data flowing:

A screenshot of a computer

AI-generated content may be incorrect.

**Troubleshooting**

If adding the DB2 stage causes an error on your engine, follow these steps:

1 – Access the machine where your engine is installed.  
2 – Navigate to:

cd /home/<user>/sdc/streamsets-datacollector-6.2.1/etc

3 – Open the file:

vi sdc.properties

4 – Look for the flight service property and change:

A screenshot of a computer program

AI-generated content may be incorrect.

5 – Restart the Engine.