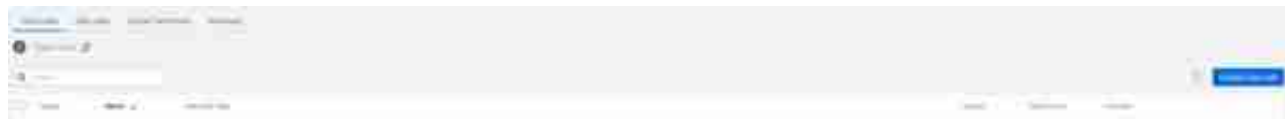


CDF (Cloudera 数据流) 和 CSA Cloudera Streaming Analytics 中提供了所有注释组件：

CLOUDERA 动态数据平台



Data Hub : 7.2.14 -使用 Apache NiFi、Apache NiFi Registry 的轻型流量管理



Data Hub : 7.2.14 -使用 Apache Flink 进行轻型流分析



让我们开始在 NiFi 中获取我们的数据。使用 InvokeHTTP Processor，我们可以从randomuser API 收集所有数据。

Configure Processor | JoltTransformJSON 1.15.2.2.4.0-143

Stopped

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field 🔍 +

Property	Value
Jolt Transformation DSL	? Chain ↑
Custom Transformation Class Name	? No value set ↑
Custom Module Directory	? No value set ↑
Jolt Configuration	? [{"operation": "shift", "spec": {"results": {"*": {"login": {"username": "customer_id", "uuid": "account_number"}, "name": {"first": "name", "last": "lastname"}, "email": "email", "gender": "gender", "location": {"street": {"number": "charge_amount"}, "country": "country", "state": "state", "city": "city", "coordinates": {"latitude": "lat", "longitude": "lon"}}, "picture": {"large": "image"}}}}}, {"operation": "default", "spec": {"center_inferred_lat": -5.0000, "center_inferred_lon": -5.0000, "max_inferred_distance": 0.0, "max_inferred_amount": 0.0}}, {"operation": "modify-overwrite-beta", "spec": {"lat": "=toDouble", "lon": "=toDouble"}}}] ↑

我们将使用JOLT转换来清理和调整我们的数据：

```
[{"operation": "shift", "spec": {"results": {"*": {"login": {"username": "customer_id", "uuid": "account_number"}, "name": {"first": "name", "last": "lastname"}, "email": "email", "gender": "gender", "location": {"street": {"number": "charge_amount"}, "country": "country", "state": "state", "city": "city", "coordinates": {"latitude": "lat", "longitude": "lon"}}, "picture": {"large": "image"}}}}}, {"operation": "default", "spec": {"center_inferred_lat": -5.0000, "center_inferred_lon": -5.0000, "max_inferred_distance": 0.0, "max_inferred_amount": 0.0}}, {"operation": "modify-overwrite-beta", "spec": {"lat": "=toDouble", "lon": "=toDouble"}}}]
```

我们的输出转换数据将是：

```
Result:{"customer_id" : "organicfrog175","account_number" : "d73f9a11-d61c-424d-8309-51d6d8e83a73","name" : "Shirlei","lastname" : "Freitas","email" : "shirlei.freitas@example.com","gender" : "female","charge_amount" : 6133,"country" : "Brazil","state" : "Amapá","city" : "Belford Roxo","lat" : 78.0376,"lon" : 74.2175,"image" : "https://randomuser.me/api/portraits/women/82.jpg","max_inferred_distance" : 0.0,"center_inferred_lat" : -5.0,"center_inferred_lon" : -5.0,"max_inferred_amount" : 0.0}
```

现在，我们可以使用UpdateRecord 处理器来改进它并在某些字段中获取一些随机数，因此，使用PublishKafka2RecordCDP处理器将我们的 JSON 数据放入 Kafka。

更新记录处理器

Configure Processor | PublishKafka2RecordCDP 1.0.0.2.2.4.0-143

Stopped

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value
Kafka Brokers	#{Kafka Broker Endpoint}
Topic Name	#{Kafka Destination Topic}
Record Reader	simple-JsonTreeReader
Record Writer	simple-JsonRecordSetWriter
Use Transactions	false
Failure Strategy	Route to Failure
Delivery Guarantee	Guarantee Single Node Delivery
Attributes to Send as Headers (Regex)	No value set
Message Header Encoding	UTF-8
Security Protocol	SASL_SSL
SASL Mechanism	PLAIN
Kerberos Credentials Service	No value set

(重要的是要注意必须根据 Kafka 集群端点填充的 Kafka 代理变量。)

最后，我们的 NiFi 流程将是这样的：

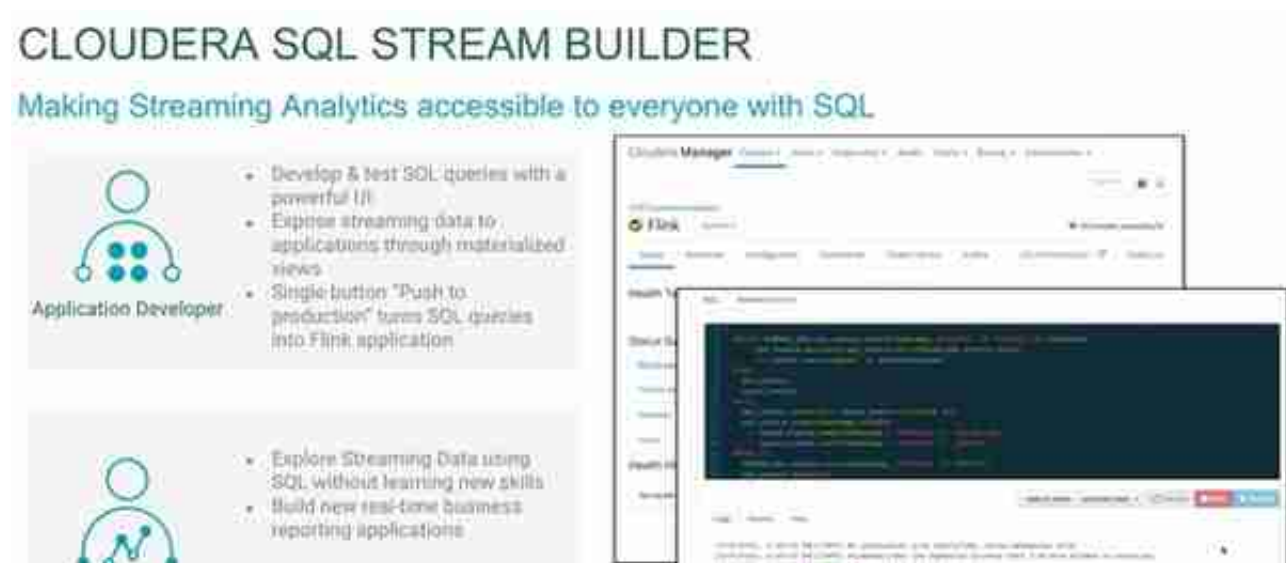


在 Kafka 集群上，我们只需点击 SMM（流消息管理器）组件中的“添加新”按钮即可创建一个新的 Kafka 主题：我已经创建了 skilltransactions 作为示例。



查看到目前为止所有摄取的数据。

流式 SQL 分析



我们将在 SSB 上的表连接器上轻松创建我们的“虚拟表”映射：



创建这个“虚拟表”后，我们可以使用 SQL 对使用 power、sin 和 radians SQL 函数进行的交易进行了多远的数学计算：

```
select account_number, charge_amount, 2 * 3961 * asin(sqrt(power(power((sin(radians((lat - center_inferred_lat) / 2))), 2) + cos(radians(center_inferred_lat)) * cos(radians(lat)) * (sin(radians((lon - center_inferred_lon) / 2))), 2))) as distance, max_inferred_distance, max_inferred_amount from `skilletransactions` WHERE 2 * 3961 * asin(sqrt(power(power((sin(radians((lat - center_inferred_lat) / 2))), 2) + cos(radians(center_inferred_lat)) * cos(radians(lat)) * (sin(radians((lon - center_inferred_lon) / 2))), 2))) > max_inferred_distance
```

要查看有关此查询的更多详细信息，请访问我们 Cloudera 社区上 @sunile_manjee 撰写的这篇精彩文章。

我们还可以创建我们的函数，然后调用它或查询。

例如，让我们创建一个 `DISTANCE_BETWEEN` 函数并在我们的最终查询中使用它。

最终查询

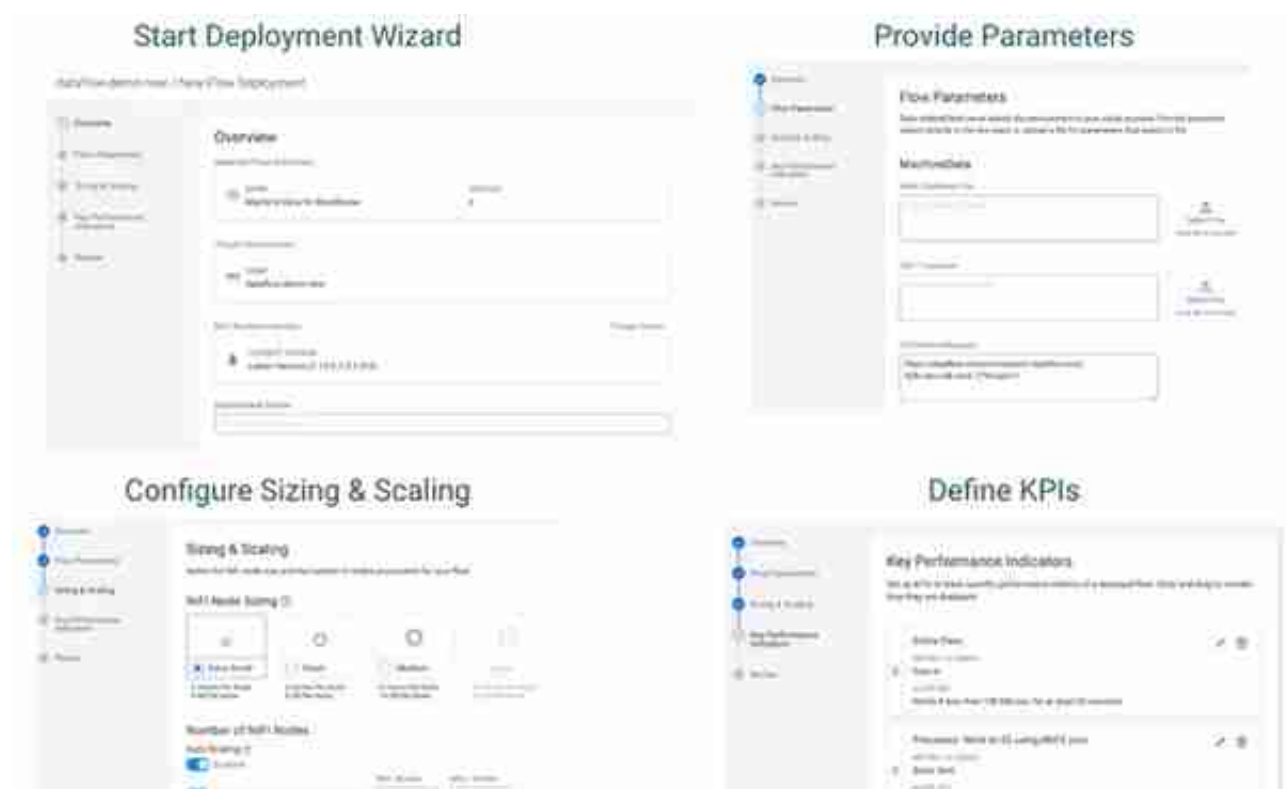
```
select account_number, charge_amount, DISTANCE_BETWEEN(lat,
lon, center_inferred_lat, center_inferred_lon) as distance,
max_inferred_distance, max_inferred_amount from `skilltransactions`
WHERE DISTANCE_BETWEEN(lat, lon, center_inferred_lat,
center_inferred_lon) > max_inferred_distance OR charge_amount
> max_inferred_amount
```

此时我们的查询应该可以实时检测到可疑交易，可以报警了。

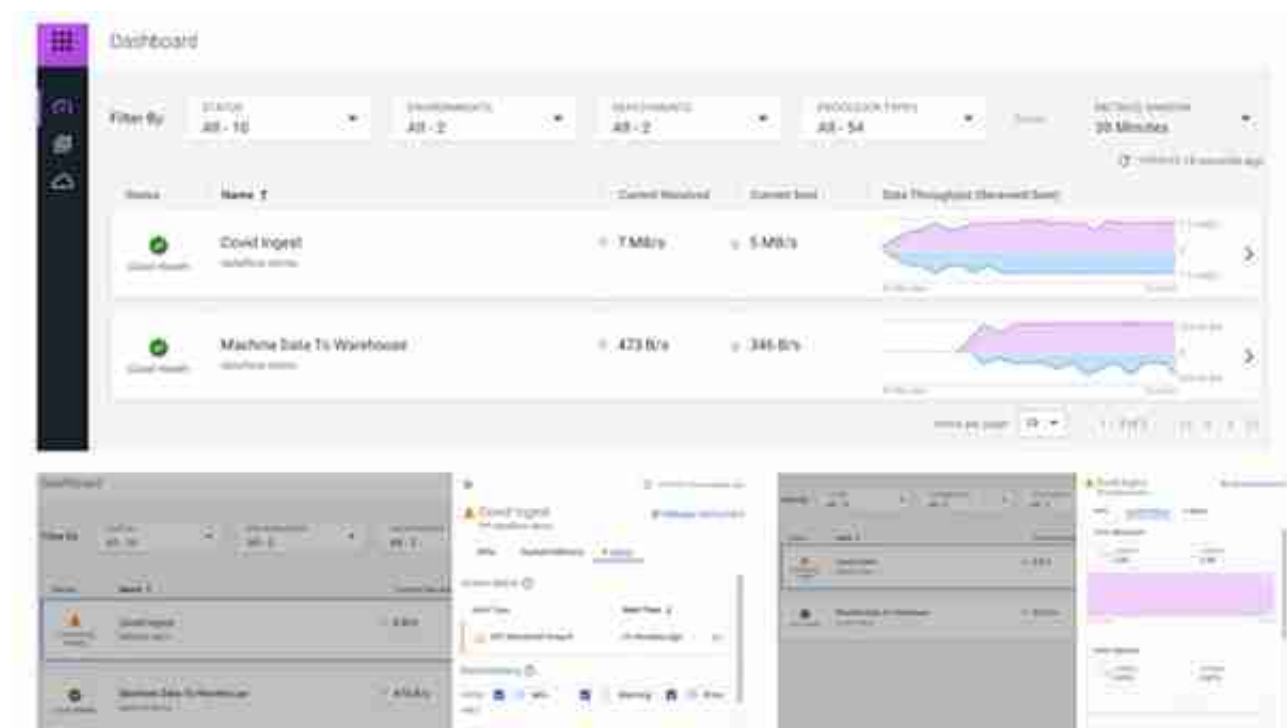


Cloudera DataFlow 服务可以在 Kubernetes 中部署 NiFi 流，提供生产环境所需的所有可扩展性。

CLOUDERA 数据流服务——公有云



关键绩效指标



部署管理器



原文作者：ThiagoSantiago

原文链接：<https://community.cloudera.com/t5/Community-Articles/Simple-Credit-Card-Fraud-Detection-with-NiFi-Kafka-Flink-and/ta-p/340228>