

Tarea módulo data streaming.

Objetivo

El objetivo de esta tarea es fijar los contenidos desarrollados durante el módulo de Data Streaming, incluyendo la aplicación de herramientas como Redpanda y KSQLDB para la solución de un problema.

Profesor:

Rodrigo Parra, M.Sc.

Grupo 3

- ❖ Jonathan Bernal
- ❖ Micheli De Almeida Bareiro
- ❖ Lourdes Beatriz Delgado Gonzalez
- ❖ Vladimir Lopez Maldonado
- ❖ Edgardo Daniel Meza Fleitas



Tarea módulo data streaming

Add comment

Objetivo

El objetivo de esta tarea es fijar los contenidos desarrollados durante el módulo de Data Streaming, incluyendo la aplicación de herramientas como Redpanda y KSQLDB para la solución de un problema.

Instrucciones

La tarea consiste en construir un servicio que consuma la API en tiempo real de Finnhub: <https://finnhub.io/docs/api/websocket-trades> y consuma actualizaciones para los siguientes símbolos:

- AAPL
- AMZN
- BINANCE:BTCUSDT

De modo a procesar las actualizaciones, deben seguirse los siguientes pasos:

1. Instalar un cluster Redpanda de manera local utilizando un archivo *docker-compose.yml*
2. Implementar un producer utilizando [kafka-python](#), de acuerdo a la [documentación](#) de Redpanda y similar al ejemplo desarrollado en clase, para suscribirse a los eventos de la API.
3. Instalar [KSQLDB](#) modificando el archivo *docker-compose.yml*, de acuerdo a la [documentación](#) de Redpanda.
4. Ejecutar ksqldb-cli, definir los *streams* y *tablas* necesarios para responder a las siguientes preguntas:
 1. ¿Cuál fue el promedio ponderado de precio de una unidad por cada uno de los símbolos procesados? (e.j. AAPL)
 2. ¿Cuántas transacciones se procesaron por símbolo?
 3. ¿Cuál fue el máximo precio registrado por símbolo?
 4. ¿Cuál fue el mínimo precio registrado por símbolo?

Presentación

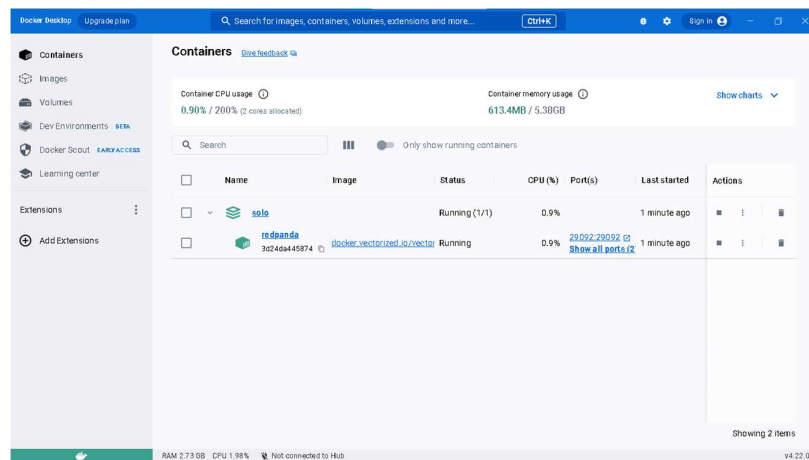
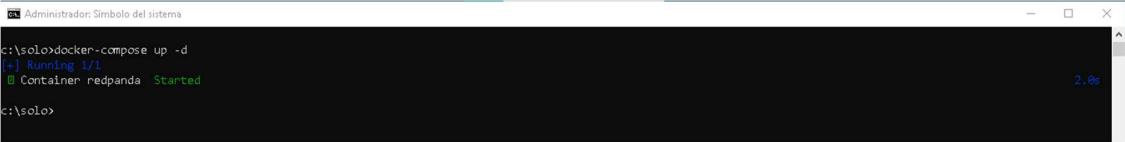
El proyecto debe desarrollarse en los grupos previamente formados. El entregable debe incluir:

- Enlace a repositorio público de Github incluyendo:
- docker-compose.yml
- código fuente del consumer
- archivos .sql con las 4 consultas utilizadas para responder las preguntas de la sección anterior
- Una grabación de pantalla mostrando el sistema en funcionamiento y explicando brevemente los pasos que se siguieron para su implementación.

Fecha límite de entrega: 21-08-2023 23:59

1. Instalar un cluster Redpanda de manera local utilizando un archivo *docker-compose.yml*

```
version: '3.9'
```



2. Implementar un producer utilizando [kafka-python](#), de acuerdo a la [documentación](#) de Redpanda y similar al ejemplo desarrollado en clase, para suscribirse a los eventos de la API.

```
Administrador: Símbolo del sistema


c:\tp>C:\Users\copaco\anaconda3\envs\p2\python c:\tp\producer_p2.py
--- request header ---
GET /?token=cj8gvvpr0lqjjsj7cm10cj8gvvpr0lqjjsj7cm1g HTTP/1.1
Upgrade: websocket
Host: ws.finnhub.io
Origin: https://ws.finnhub.io
Sec-WebSocket-Key: G9HE1vLnTuWvuMp0sZfyQ==
Sec-WebSocket-Version: 13
Connection: Upgrade

-----
--- response header ---
HTTP/1.1 101 Switching Protocols
Date: Wed, 16 Aug 2023 11:41:23 GMT
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: HfjsDV7P2RwL7mtY71DGGobh6o=
CF-Cache-Status: DYNAMIC
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=yUveaN6rTU6g7t87rogVSjY8EUjX2FUx2BvXQ5kfjNjX8SKrYGL5ZrNFCx2FztHAX2BsUjGk2F3p0uy1lEsNgqCjUfNlNvgf8bDX2BjB7pZk2BmxAAmbQu927J4dWfz2FcXtFybG9vRvREv"}],"group":"cf-nel","max_age":604800}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Server: cloudflare
CF-RAY: 7f79686a983a0325-MIA
alt-svc: h3=":443"; ma=86400

-----
Websocket connected
++Sent raw: b'\x81\x4a\x90CEU\x04\x1a\x12\x2Y\x1c\x04\x05\x01M\x14\x02n\x12\x20\x1c\x04\x09\x0e\x18\xfcA\x04U\x01n\x02\x1e'
++Sent decoded: fin=1 opcode=1 data=b'{"type":"subscribe","symbol":"AAPL"}'
++Sent raw: b'\x81\x4a\x0b\x09\xef\x00\xeb\x96\xfb\xac\r\x05\x0a\x0b\x0d\x08\x0a\x06\x09\xac\r\x03\x0a9\x0bA\x02\x09\x06C\x0d\x0b1\x0ebn\x02\x01\x07\r\x92'
++Sent decoded: fin=1 opcode=1 data=b'{"type":"subscribe","symbol":"AMZN"}'
++Sent raw: b'\x81\xafD\x8f\x17\x9b\x0d\x0e24\x0a5\x0a1f\xfb\x97\x0e\x02\x0a5\x0b7f\xfc\r\x06\x0e0\x0b9\x0adU\x02n\x0ceV\x08\x01\x0b5U\x0cf\x07\x0daD\x0d\x0a\x0ad'
++Sent decoded: fin=1 opcode=1 data=b'{"type":"subscribe","symbol":"BINANCE:BTCUSD"}'
++Rcv raw: b'\x81e("data":[{"c":null,"p":29159.08,"s":"BINANCE:BTCUSD","t":1692186082944,"v":0.00199}],{"type":"trade"}'
++Rcv decoded: fin=1 opcode=1 data=b'{"data":[{"c":null,"p":29159.08,"s":"BINANCE:BTCUSD","t":1692186082944,"v":0.00199}],{"type":"trade"}'
Message produced to topic 'stock-updates' at offset 0
++Rcv raw: b'\x81-\x01H("data":[{"c":null,"p":29159.08,"s":"BINANCE:BTCUSD","t":1692186083248,"v":0.00855}, {"c":null,"p":29159.09,"s":"BINANCE:BTCUSD","t":1692186083289,"v":0.0006}, {"c":null,"p":29159.08,"s":"BINANCE:BTCUSD","t":1692186084182,"v":0.00216}, {"c":null,"p":29159.09,"s":"BINANCE:BTCUSD","t":1692186084195,"v":0.00329}],{"type":"trade"}'
++Rcv decoded: fin=1 opcode=1 data=b'{"data":[{"c":null,"p":29159.08,"s":"BINANCE:BTCUSD","t":1692186083248,"v":0.00855}, {"c":null,"p":29159.09,"s":":B
```

[illegible]

3. Instalar [KSQLEDB](#) modificando el archivo `docker-compose.yml`, de acuerdo a la [documentación](#) de Redpanda.



```
Administrador: Símbolo del sistema - docker compose up -d

c:\tp>
c:\tp>docker compose up -d
[+] Running 4/5
  Network tp_default          Created           0.3s
  - Container redpanda        Starting         5.4s
  Container redpanda-console  Created          2.1s
  Container ksqldb-server     Created          1.7s
  Container ksqldb-cli        Created          0.3s
```

Desktop

Upgrade plan

Search for images, containers, volumes, extensions and more...

Ctrl+K

Sign in

Containers

Images

Volumes

Dev Environments BETA

Docker Scout EARLY ACCESS

Learning center

Extensions

Add Extensions

Containers

Container CPU usage 17.83% / 200% (2 cores allocated)

Container memory usage 1.39GB / 5.38GB

Show charts

Search

Only show running containers

	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	<div><div>tp</div></div>		Running (4/4)	17.83%		43 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div><div>redpanda</div><div>b2a52e805b3a</div></div>	docker.vectorized.io/vector	Running	14.92%	29092:29092 Show all ports (3)	43 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div><div>ksqldb-server</div><div>6007905f62cd</div></div>	confluentinc/ksqldb-server	Running	2.75%	8088:8088	43 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div><div>ksqldb-cli</div><div>47ab0a98cf43</div></div>	confluentinc/ksqldb-cli:0.2	Running	0.16%		43 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div><div>redpanda-console</div><div>a02b91804e7f</div></div>	docker.redpanda.com/vect	Running	0%	8080:8080	43 minutes ago	<div></div> <div></div> <div></div>

Showing 5 items

Notepad

4. Ejecutar `ksqldb-cli`, definir los *streams* y *tablas* necesarios para responder a las siguientes preguntas:

```
c:\>docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
```

```
=====
=                                     =
=      [K] [S] [Q] [L] [D]          =
=      [B]                          =
=                                     =
=      The Database purpose-built    =
=      for stream processing apps    =
=====
```

```
Copyright 2017-2021 Confluent Inc.

CLI v0.24.0, Server v0.24.0 located at http://ksqldb-server:8088
Server Status: RUNNING

Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!

ksql> CREATE STREAM tp2 (symbol VARCHAR, price DOUBLE, volume DOUBLE)
> WITH (kafka_topic='stock-updates', value_format='json', partitions=1);

Message
-----
Stream created
-----

ksql>
```

```

C:\Adminstrador: Símbolo del sistema - docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
c:\tp>docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.

=====
|                                     |
|  KSQLDB                            |
|                                     |
|  The Database purpose-built        |
|  for stream processing apps       |
|                                     |
=====

Copyright 2017-2021 Confluent Inc.

CLI v0.24.0, Server v0.24.0 located at http://ksqldb-server:8088
Server Status: RUNNING

Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!

ksql> CREATE STREAM tp2 (symbol VARCHAR,price DOUBLE, volume DOUBLE)
> WITH (kafka_topic='stock-updates', value_format='json', partitions=1);

Message
-----
Stream created
-----

ksql> CREATE TABLE promedio AS
> SELECT symbol, ( sum(price * volume) / SUM(volume) ) AS aveedage , count(symbol) as transacciones FROM tp2
> GROUP BY symbol
> EMIT CHANGES;

Message
-----
Created query with ID CTAS_PROMEDIO_6
-----

ksql>
ksql> CREATE TABLE mima AS
> SELECT symbol, max(price) AS maximo, min(volume) AS minimo FROM tp2
> GROUP BY symbol
> EMIT CHANGES;

Message
-----
Created query with ID CTAS_MIMA_10
-----

ksql>

```

1. ¿Cuál fue el promedio ponderado de precio de una unidad por cada uno de los símbolos procesados? (e.j. AAPL)

```
Simbolo del sistema - docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!

ksql> select symbol, aveerdage from promedio;
+-----+-----+
|SYMBOL|AVEERDAGE|
+-----+-----+
|AAPL  |177.69956834532374|
|AMZN  |137.80479889042996|
|BINANCE:BTCUSD|29158.66369146906|
Query terminated
ksql>
```

2. ¿Cuántas transacciones se procesaron por símbolo?

```
Simbolo del sistema - docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
|BINANCE:BTCUSD|29158.66369146906|
Query terminated
ksql> select symbol, transacciones from promedio;
+-----+-----+
|SYMBOL|TRANSACCIONES|
+-----+-----+
|AAPL  |4|
|AMZN  |18|
|BINANCE:BTCUSD|436|
Query terminated
ksql>
```

3. ¿Cuál fue el máximo precio registrado por símbolo?

```
Simbolo del sistema - docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
|BINANCE:BTCUSD|436|
Query terminated
ksql> select symbol, maximo from mima;
+-----+-----+
|SYMBOL|MAXIMO|
+-----+-----+
|AAPL  |177.7|
|AMZN  |137.82|
|BINANCE:BTCUSD|29159.09|
Query terminated
ksql>
```

4. ¿Cuál fue el mínimo precio registrado por símbolo?

```
Simbolo del sistema - docker exec -it ksqldb-cli ksql http://ksqldb-server:8088
|BINANCE:BTCUSD|29159.09|
Query terminated
ksql> select symbol, minimo from mima;
+-----+-----+
|SYMBOL|MINIMO|
+-----+-----+
|AAPL  |1.0|
|AMZN  |1.0|
|BINANCE:BTCUSD|0.00034|
Query terminated
ksql>
```