Batch

Egy korábbi projectmunka csv formátumú filet feltöltöttem databricksre és táblává alakítom.

Table

Description automatically generated

A létrehozott tábla így néz ki:

Table

Description automatically generated

Ezek után csinálok egy DataFrame-et amibe generálok egy új oszlopot random számokkal 0-12 között.

Graphical user interface, application, table

Description automatically generated

Ezek után létrehozok egy új táblát a service id-t kihagyva, de egy új time BIGINT oszloppal kiegészítve.

Graphical user interface, text, application

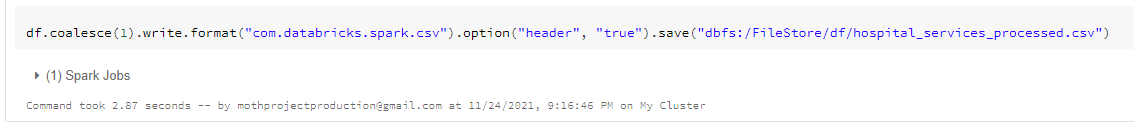
Description automatically generated

Ezek után a DataFrame-t beírom ebbe az új táblába, amit utána megjelenítek és leellenőrzöm a tartalmát.

Table

Description automatically generated

Végül az új tábla tartalmát kiírom egy csv fileba.

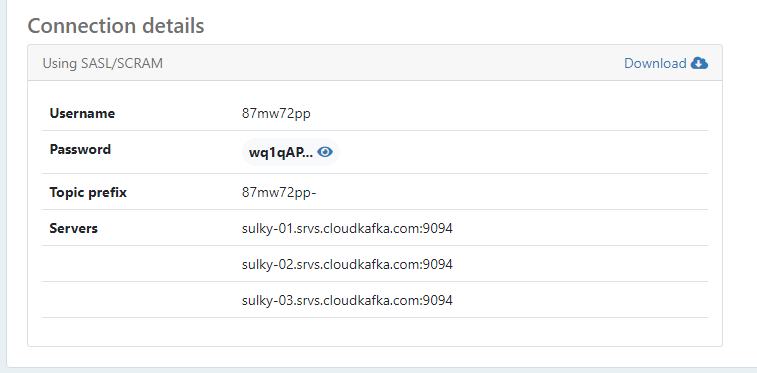


Streaming

Karafkán belül létre kell hozni egy új Instance-ot.

Graphical user interface, text, application

Description automatically generated  
A létrejött instance Details tabján található adatok alapján fel lehet építeni a kapcsolatot a Producer és Consumer számára a DataBricks projecten belül.





A Produceren belül a kriptovaluta árfolyamokból a következő adatokat szedtem ki és kezdtem el kafkára streamelni: 'id', 'rank', 'symbol', 'name', 'priceUsd', 'changePercent24Hr'. Az elindított Producer által elindított stream látható a karafka oldalon a Browser menüpont alatt.

Graphical user interface, text, application

Description automatically generated

A tesztelés alapján működik, így be lehet kötni a Consumert is. A korábbi konfiguráció alapján beállítjuk a megfelelő username, jelszó, server párosításokat és megadjuk, hogy melyik porton legyenek elérhetők az adatok amik átfolynak rajta.

Text

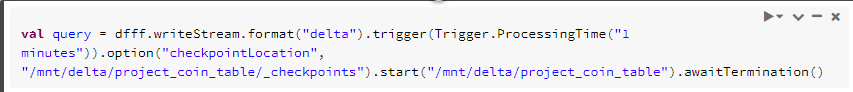
Description automatically generated

Ha minden jól megy, akkor alul láthatóváválnak az üzenetek amik azt jelzik, hogy a Consumer megkapja a streamelt adatokat.

Text

Description automatically generated

Ezek után létrehozunk egy DB Processort amivel egy táblában letároljuk az adatokat amik streamen érkeznek. Ehhez készítünk egy táblát ami megfeleltethető a forrásból kiszedett adatokkal, majd a Consumerben megadott portot figyelve először feldolgozzuk egy adatstruktúrába amivel könnyebb lesz a táblába való mentés, majd elindítjuk.



Ezek az adatok meg is jelennek a korábban létrehozott táblában.

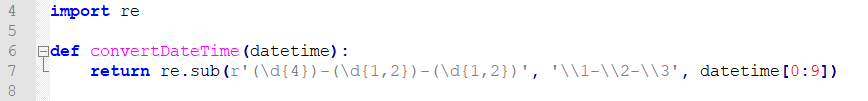
Table

Description automatically generated

XML

Egy employee adatokat tartalmazó xml filet szeretnénk feldolgozni olyan módon, hogy a benne található dátumok valamilyen könnyebben értelmezhető formátumba kerüljenek, akár még db-be is be lehessen menteni őket.

Ehhez létrehozunk egy converter függvényt amihez beimportáljuk az re csomagot:



Mivel az eredeti xml-ben egy hosszú időbélyegzett time format van, ezért le kell metszenünk az eredeti hosszból és utána más sorrendben adjuk vissza a yyyy-mm-dd helyett.



Ez után megcsináljuk a feldolgozó függvényünket amiben a korábban létrehozott átalakító függvény segítségével az ismert datetime mezőket már az új formátumunkban mentjük el.

Graphical user interface, text, application

Description automatically generated

Ezt követően pedig kimentjük a dict-ünket egy avro fileba a pandavro és a pandas segítségével.

Text

Description automatically generated

Az így keletkezett fileban már látható az átalakított formátum.

Graphical user interface, text, application, email

Description automatically generated