

Instituto Tecnológico de Costa Rica

Programa de Ciencia de los Datos - Módulo Big Data

Tarea #2

- Esteban Sáenz Villalobos (esaenz7@gmail.com)
- Entrega: 15 de agosto 2021, 23:00.
- Observaciones: Ejecutar el programa siguiendo las instrucciones detalladas a continuación.

Instrucciones

1- Para cargar el contenedor con todos los recursos necesarios, ejecute los archivos:

a- `clean_docker.sh`. Este script borrará contenedores e imágenes antiguos correspondientes a este proyecto. Atención: el comando realiza una acción de "prune" para la limpieza.

b- `build_image.sh`. Construye una imagen a partir del archivo `DockerFile`.

c- `run_image.sh`. Este script creará 2 contenedores y una red local en docker de la siguiente forma:

* Red: `bigdatanet`, IP: `10.7.84.0/24`.

* Host principal (sesión bash donde se ejecutarán los comandos): `bigdata_tarea2_esv`, IP: `10.7.84.101`.

* Host secundario (base de datos): `postgres`, IP: `10.7.84.102`.

2- Programa principal:

a- Para ejecutar el programa principal se debe aplicar el siguiente comando:

```
#spark-submit programaestudiante.py persona*.json
```

b- Para ejecutar las pruebas del programa principal se debe aplicar el siguiente comando:

```
#python -m pytest -vv test_programaestudiante.py
```

c- Para ejecutar las instrucciones 2 y 3 de forma automática, ejecute el archivo `run.sh`.

5- Parte EXTRA:

a- Para ejecutar el programa principal se debe aplicar el siguiente comando:

```
#spark-submit programaestudiante.py fpersona*.json
```

El programa reconoce los nombres `fpersona*.json` como archivos tipo JSON con la columna adicional de fecha, por lo que el mismo código es capaz de ejecutar tanto estos archivos, como los archivos originales `persona*.json` (sin la columna fecha).

b- Para ejecutar las pruebas del programa principal se debe aplicar el siguiente comando:

```
#python -m pytest -vv test_programaextra.py
```

El módulo pytest recibe como argumento el archivo `test_programaextra.py`, el cual contiene las pruebas correspondientes para los dataframes que son generados utilizando los archivos con la columna adicional de fecha.

c- Para crear la tabla de métricas dentro del contenedor de base de datos se debe aplicar el siguiente comando:

```
#PGPASSWORD=testPassword psql -h 10.7.84.102 -U postgres -p 5432 < createtable_mtricas.sql
```

El comando recibe como parámetros la dirección IP y puerto correspondientes al contenedor de "postgres", junto con el archivo `createtable_mtricas.sql`, que contiene el script de SQL para la creación de la tabla "metricas".

d- Para ejecutar el programa extra que crea el dataframe y lo inserta dentro de la tabla creada en el paso anterior, ejecute el siguiente comando:

```
#spark-submit \
--driver-class-path postgresql-42.2.14.jar \
--jars postgresql-42.2.14.jar \
programaextra.py 10.7.84.102 5432 postgres testPassword
metricas fpersona*.json
```

El comando recibe como parámetros el archivo "programaextra.py", la dirección IP y puerto (10.7.84.102 5432) correspondientes al contenedor de "postgres", el usuario y password de la base de datos (postgres testPassword), la tabla en donde se realizará el almacenamiento de los datos (metricas) y los archivos "fpersona*.json" en formato JSON correspondientes a los datos. Al final de la ejecución se hace una lectura de comprobación hacia la tabla, la cual es impresa en consola.

Detalles del trabajo

* El programa consta de los siguientes archivos:

1. `procesamientodatos.py` (lógica de procesamiento).
2. `programaestudiante.py` (programa principal).
3. `confest.py` (contexto para las pruebas).
4. `test_programaestudiante.py` (ejecución de pruebas).
5. `test_programaextra.py` (ejecución de pruebas del programa extra).
6. `programaextra.py` (programa extra).

* La aplicación principal se ejecuta por etapas (stage) ejecutadas cada una por una función en específico.

1. Stage1: cargar datos.
2. Stage2: generar tablas.
3. Stage3: almacenar tablas.

* El programa reconoce los nombres `fpersona*.json` como archivos tipo JSON con la columna adicional de fecha, por lo que el mismo código es capaz de ejecutar tanto estos archivos, como los archivos originales

persona*.json (sin la columna fecha).

- * El código fuente en cada archivo cuenta con comentarios detallados que explican la lógica del programa.

- * Se incluye un jupyter notebook de Google Colab con todo el código necesario, como complemento.

- * El repositorio completo de la tarea se encuentra también en el siguiente enlace

<<https://github.com/esaenz7/bigdataclass/tree/main/tarea2>>.
