Lyrics\_Search API Task - OUTPUT

By: Nabil Khoury

**Requirements:**

* Call musixmatch API and get English songs whose lyrics contain an <search\_keyword> and their **ALBUM** was released before 01-01-2010 . Save the output in csv format. If no search input, use “car” as default search value.
* Create a docker file exposing a service that receives a keyword and uses it as an input for the same query then prints the list of songs

**Attention:**

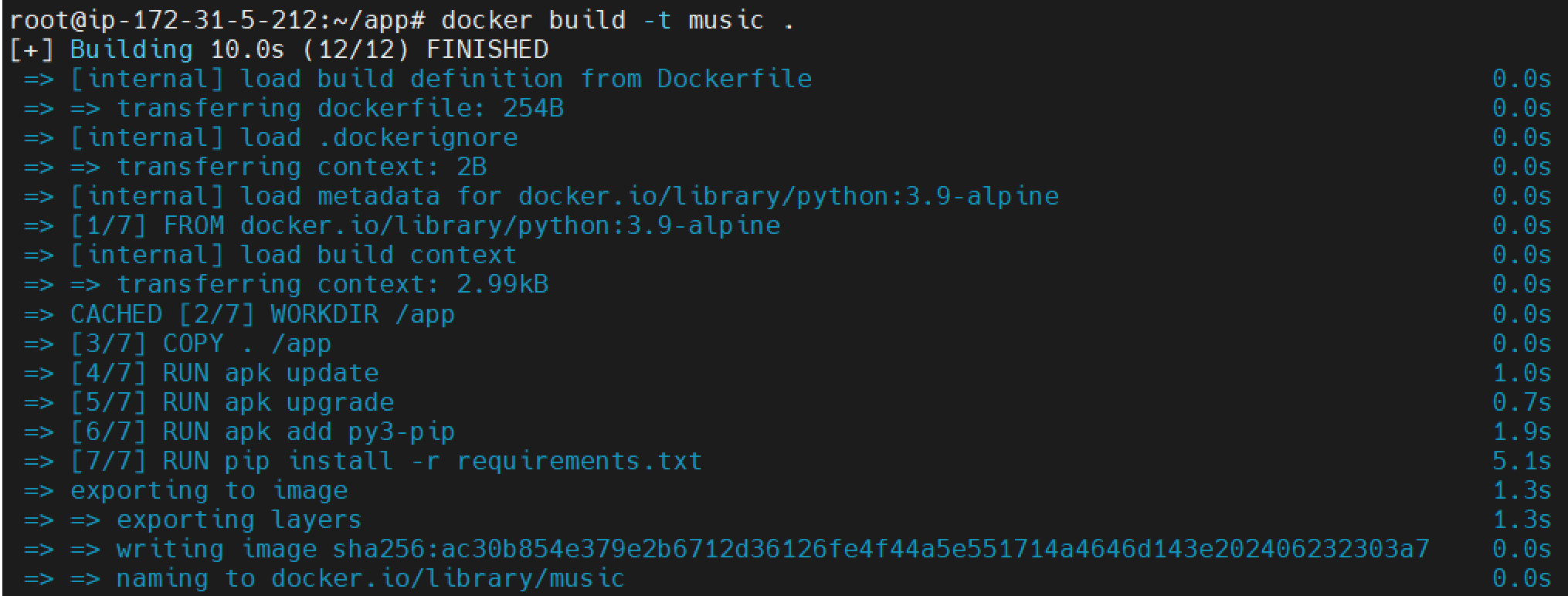
* Some albums don’t have a release date. As such, an error will be generated in the code if not handled properly. Also, other albums only have a YEAR release date which will also generate an error if not handled OR they will not be included in the result.

**PHASE 1 – Before Exposing As a Service**

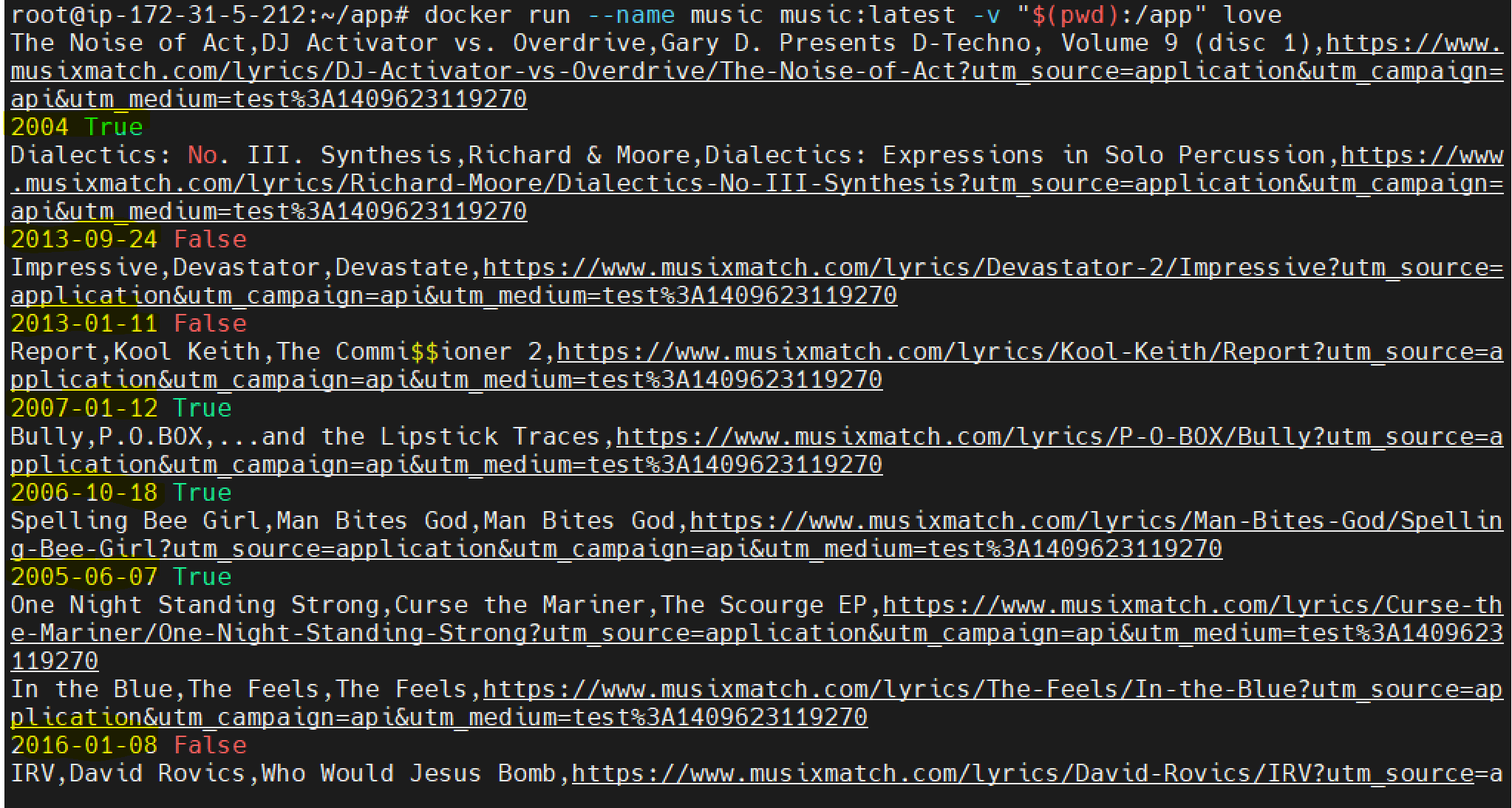
We build the image and run the script in a container. Output on screen and in csv file.

For clarity purposes, the script prints to the screen ALL songs with the SEARCH\_WORD (“love” in this case) and displays the ALBUM\_RELEASE\_DATE with True / False (whether each record meets our DATE CRITERIA), and saves the output into musixmatch.csv file.

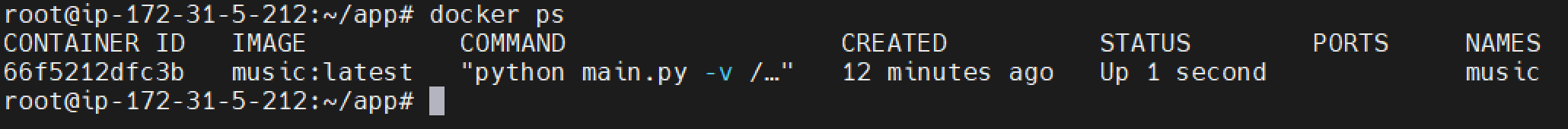
# docker build -t music .



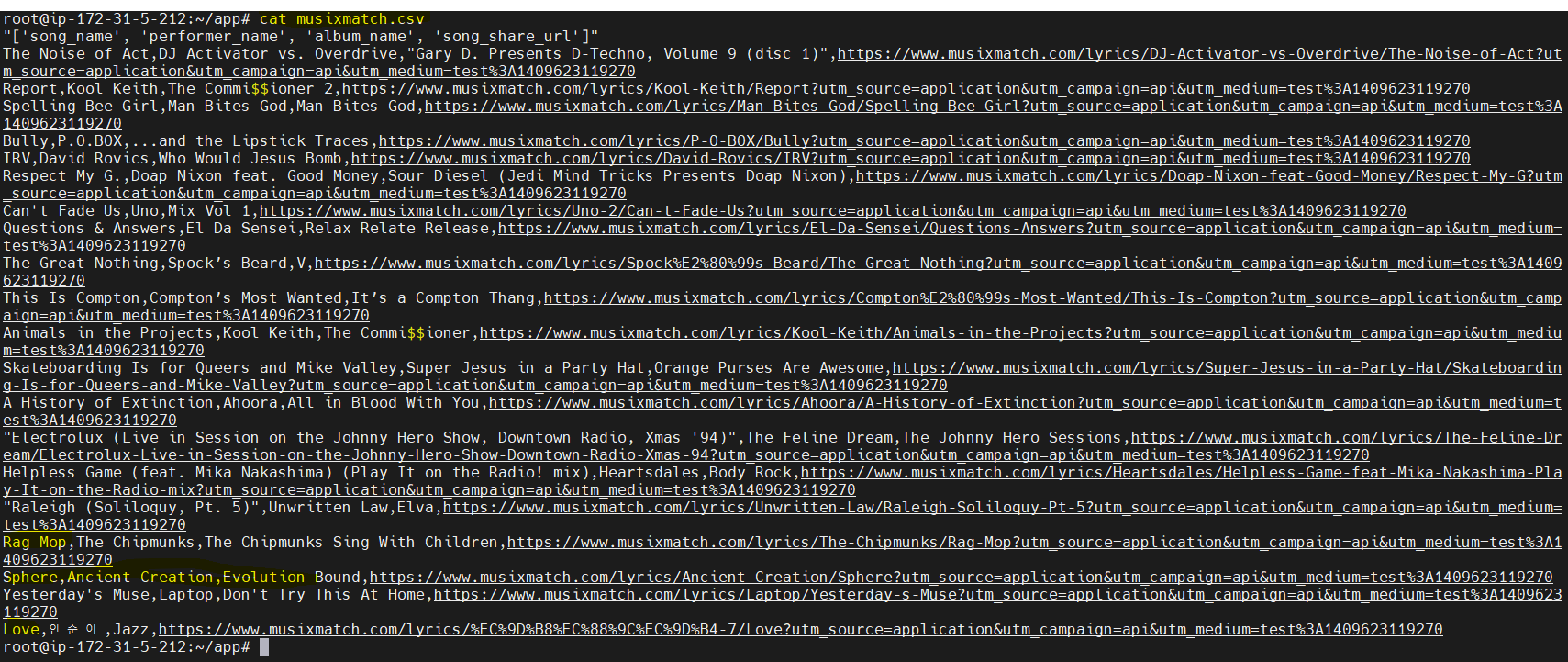
# docker run --name music music:latest -v "$(pwd):/app" love



# docker ps



# cat musixmatch.csv



**PHASE 2 – After Exposing As a Service**

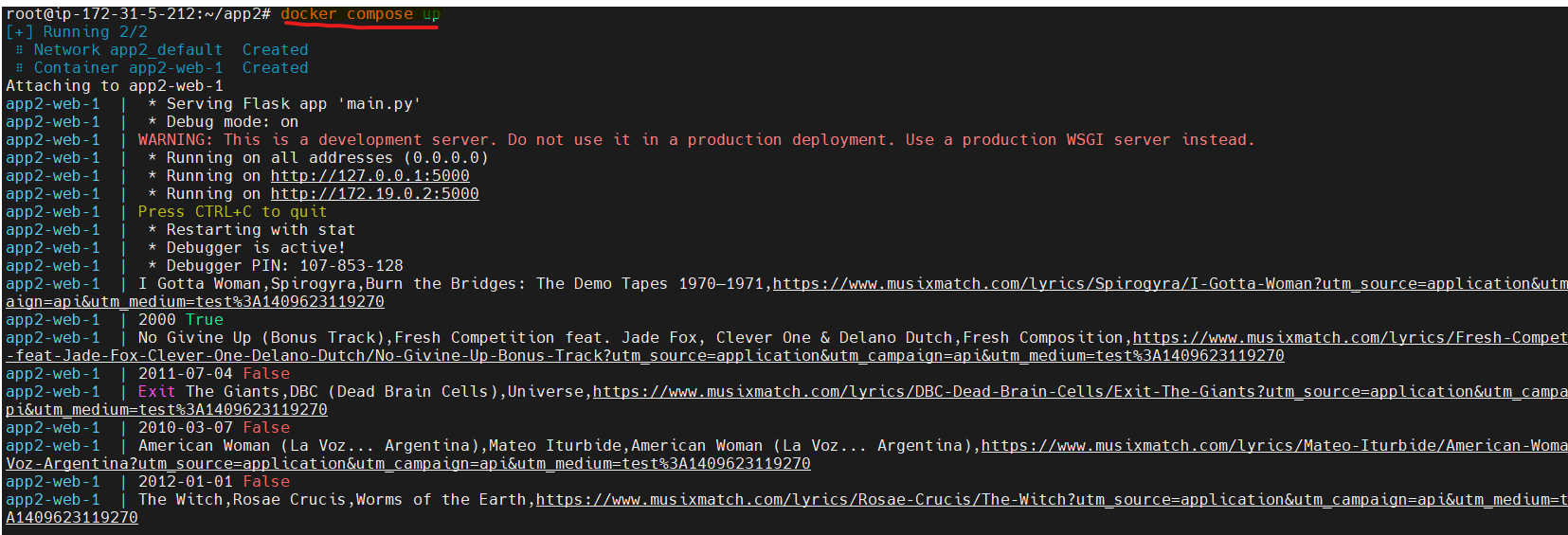
To expose the script as a service, run:

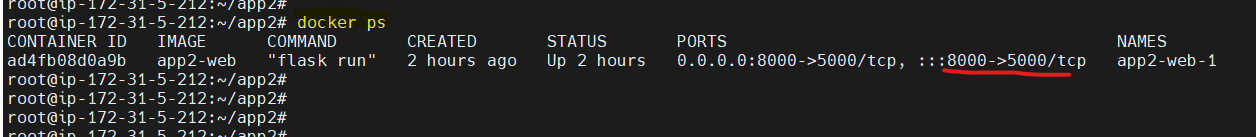
# docker compose up

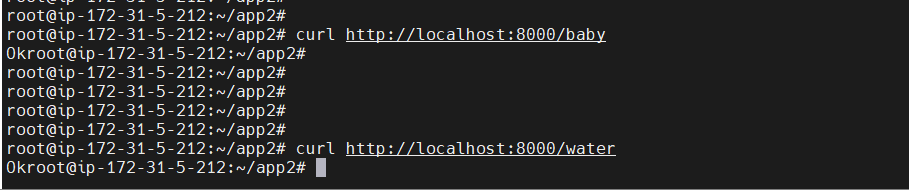
Container runs and listens on port 8000 in my case.

We can test it by opening another terminal window and typing: curl <http://localhost:8000/water>

We can watch the FIRST terminal window which will list all songs that have ‘water’ in their lyrics.







OUTPUT from 1st terminal window showing all songs with ‘water’ and 200 status\_code in the end.

Musixmatch.csv file can also be found in local directory since it has been mounted during run.

