CSV_parser

dependencies

python 3

this is the language I know best, and the one you are using as well.

postgresql

as far as I know this is a great database engine.

boto 3

the first party SDK for AWS.

psycopg

my first choice for a simple to set up python/postgresql cursor, aiohttp could be used for a more interactive service using asyncio.

data organisation

You can find the Entity-Relationship Diagram in the project doc folder.

Users are identified by their mail which is further broken down by domain name in order to simplify extraction, for instance see traffic by company.

The ‘timestamp’ is stored as a SQL date to allow for easy manipulation.

I did not use a table for the ‘ip’ field because I do not see a use for it, though it can very easily be done if needed.

The other fields have a separate table to make the data retrieval/manipulation faster.

limitations

Right now the database is very slow to build, this is because of the way the script is implemented, albeit simple, it is not efficient.

In order to overcome this limitation, and to deploy in production, we should instead retrieve the existing data from the sub tables and do most of the processing in the script. We could then send the new data using the postgresql copy function. removing all constraints and indexes then applying them back afterward could also speed up the process.

It may also be interesting to add more indexes according to the most used keys.

production

In a production setting, I would consider running the script regularly using **cron** in a linux container, it then would be very easy to set it up to send a mail if there is any issue with a run, as well as verify it is running well.