**CASE 2: Creating Data as a Service**

Team 3 in INFO 7390

Team Members: Hongwei Hu, Guangnan Liang

GitHub: <https://github.com/Besimilar/AdvancedDataAnalysis>

Docker Hub: besimilar/advanced-data-analysis:dbaas

Resources:

1. ZillowData: <https://www.kaggle.com/c/zillow-prize-1>
2. Database: AWS SimpleDB

Note: Red lines means Very Important Steps

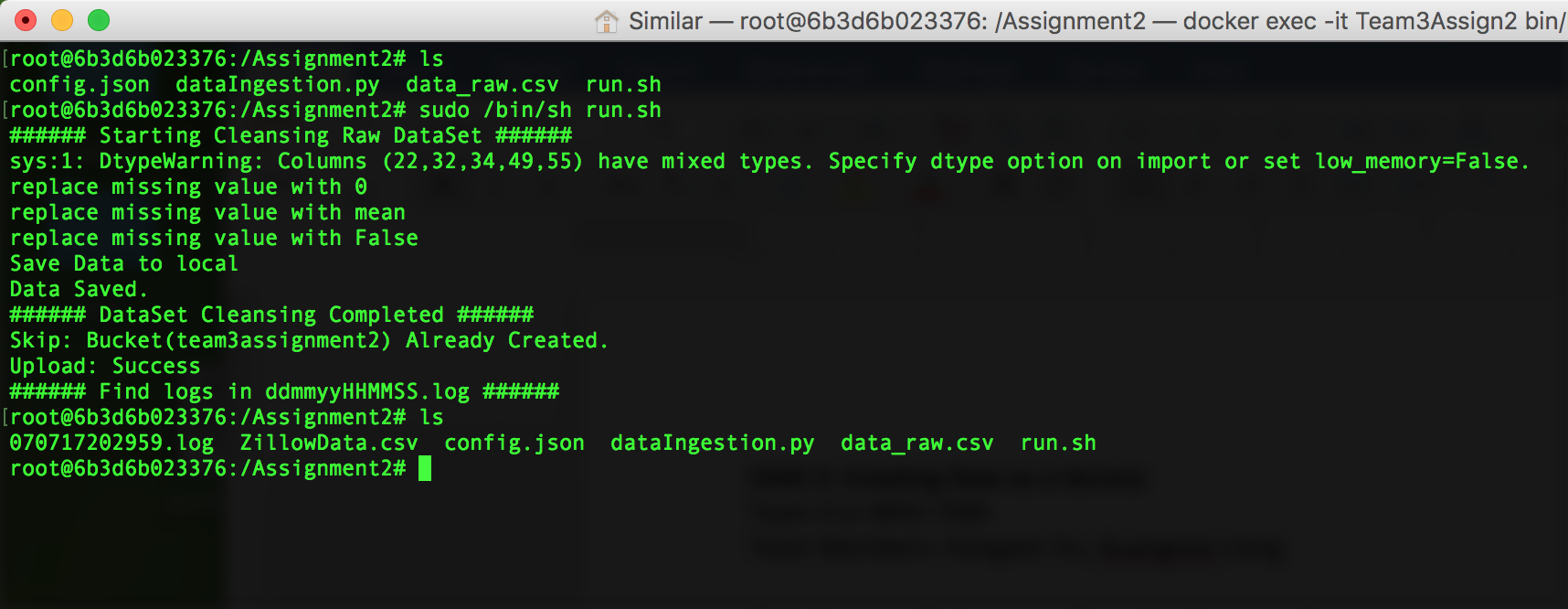
**Part1: Data Ingestion & Data Wrangling**:

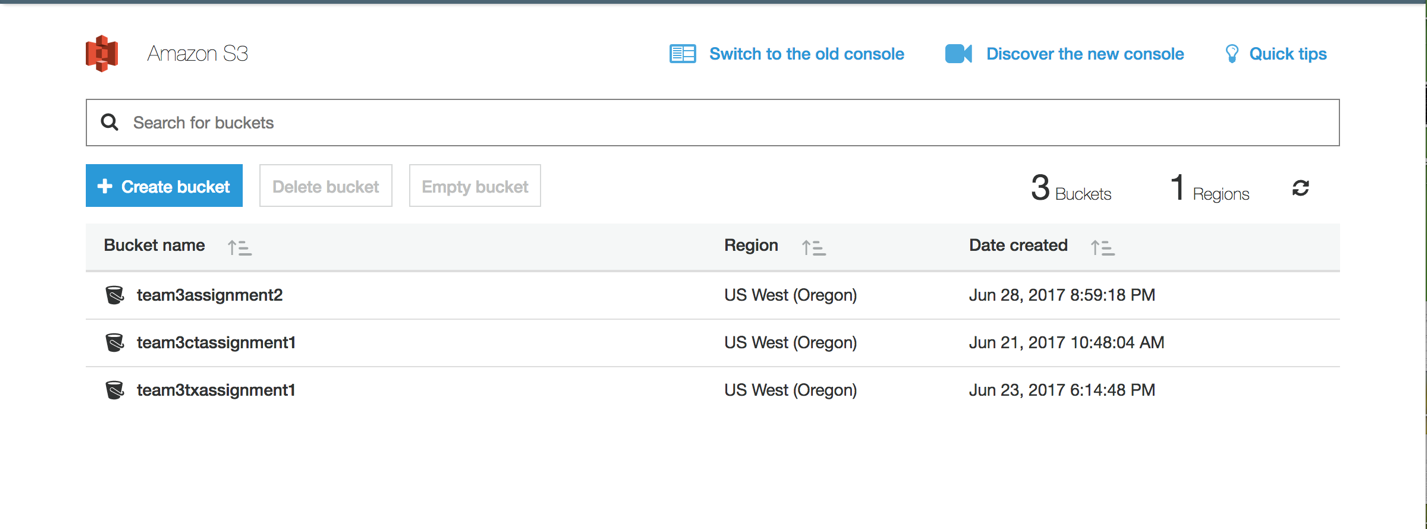
1. Configuration (Brief)

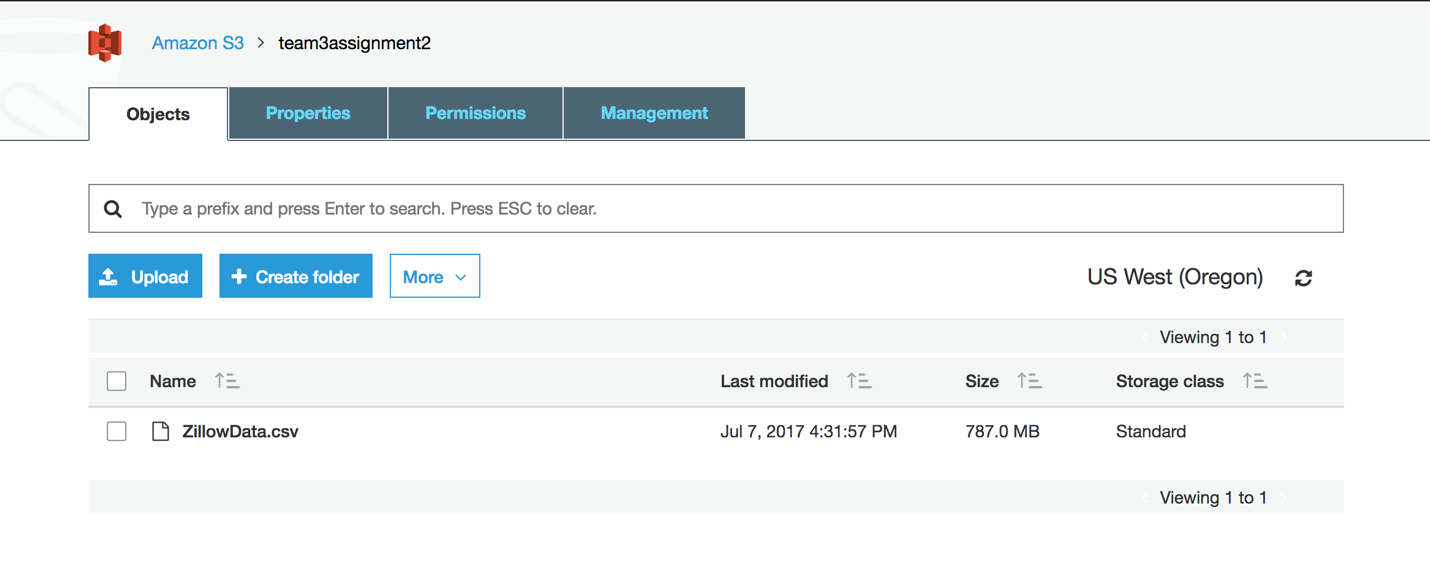
* Document: <http://boto3.readthedocs.io/en/latest/guide/migrations3.html>
* Amazon IAM Console:
* create your credentials
* download private key file
* config.json:
* modify\_config.json
  + Add AWS Keys in this file

2. dataingestion.py

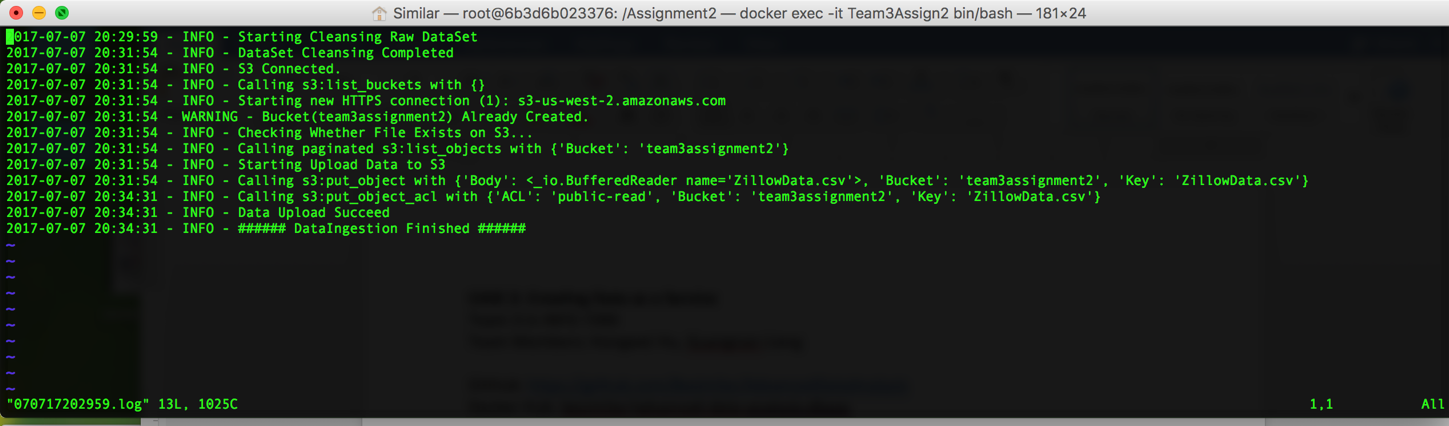
* Code: For details, please refer to code file on GitHub.
* Steps:
  1. If you have set correctly configuration files above, the Only thing you need do is:
     + $ sudo /bin/sh run.sh





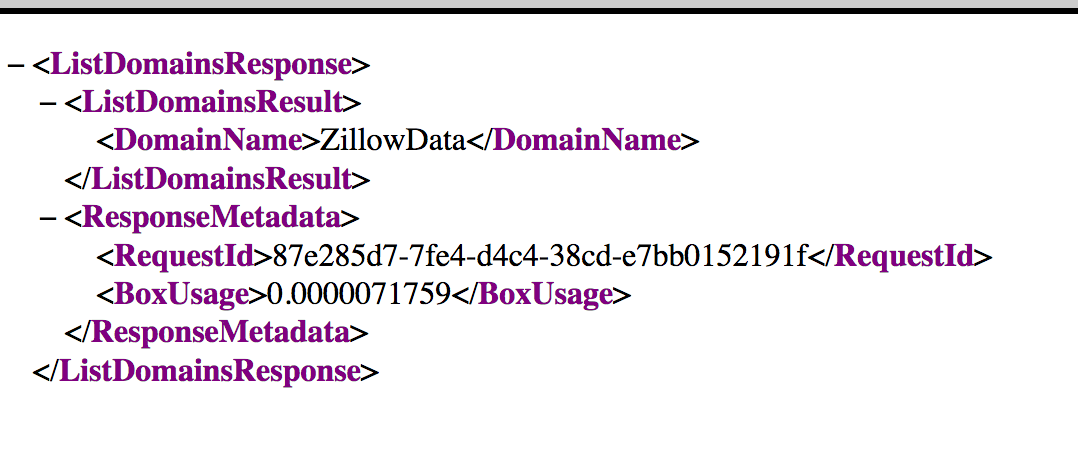


* 1. Logs into local:



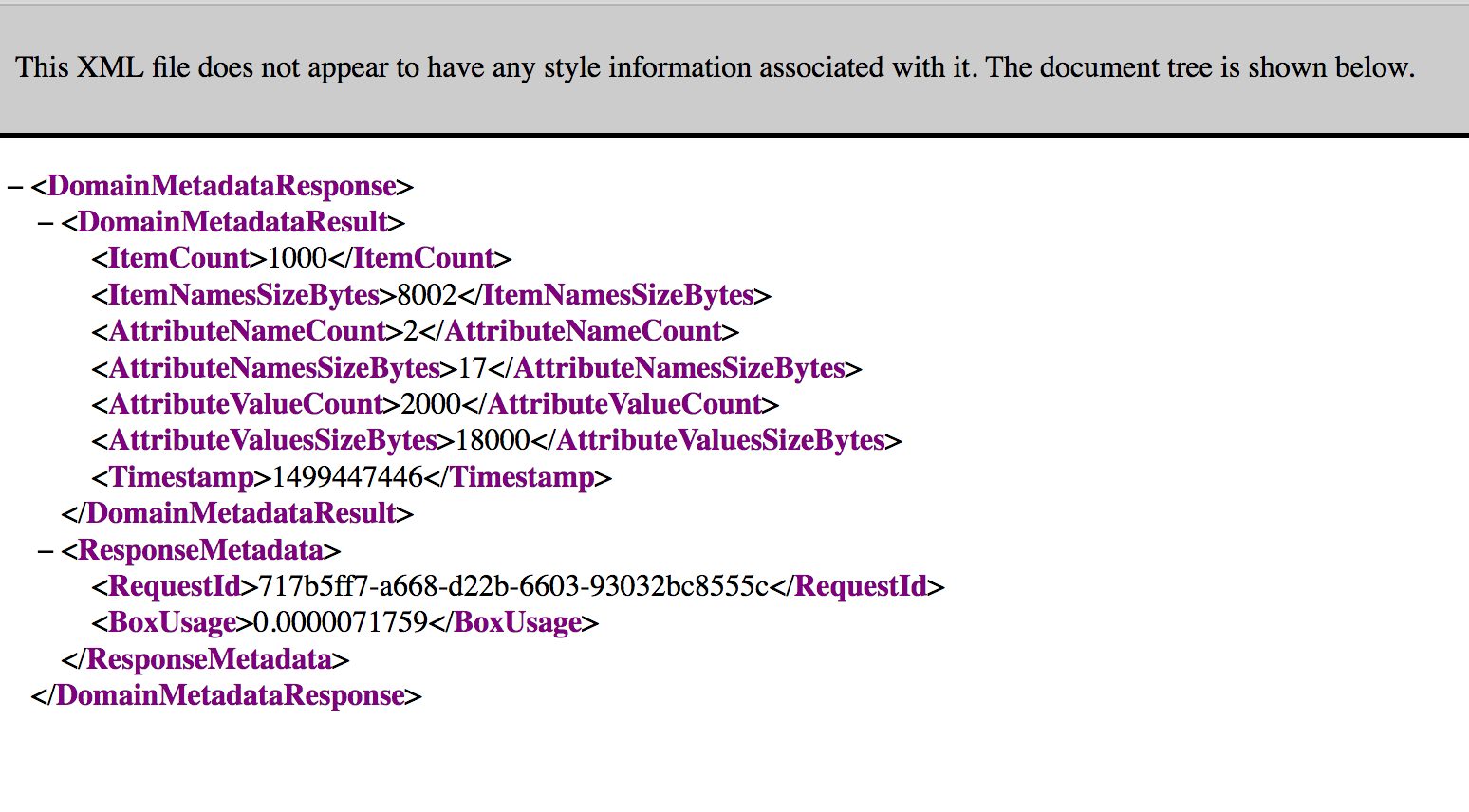
**Part2: Create a DBaas:**

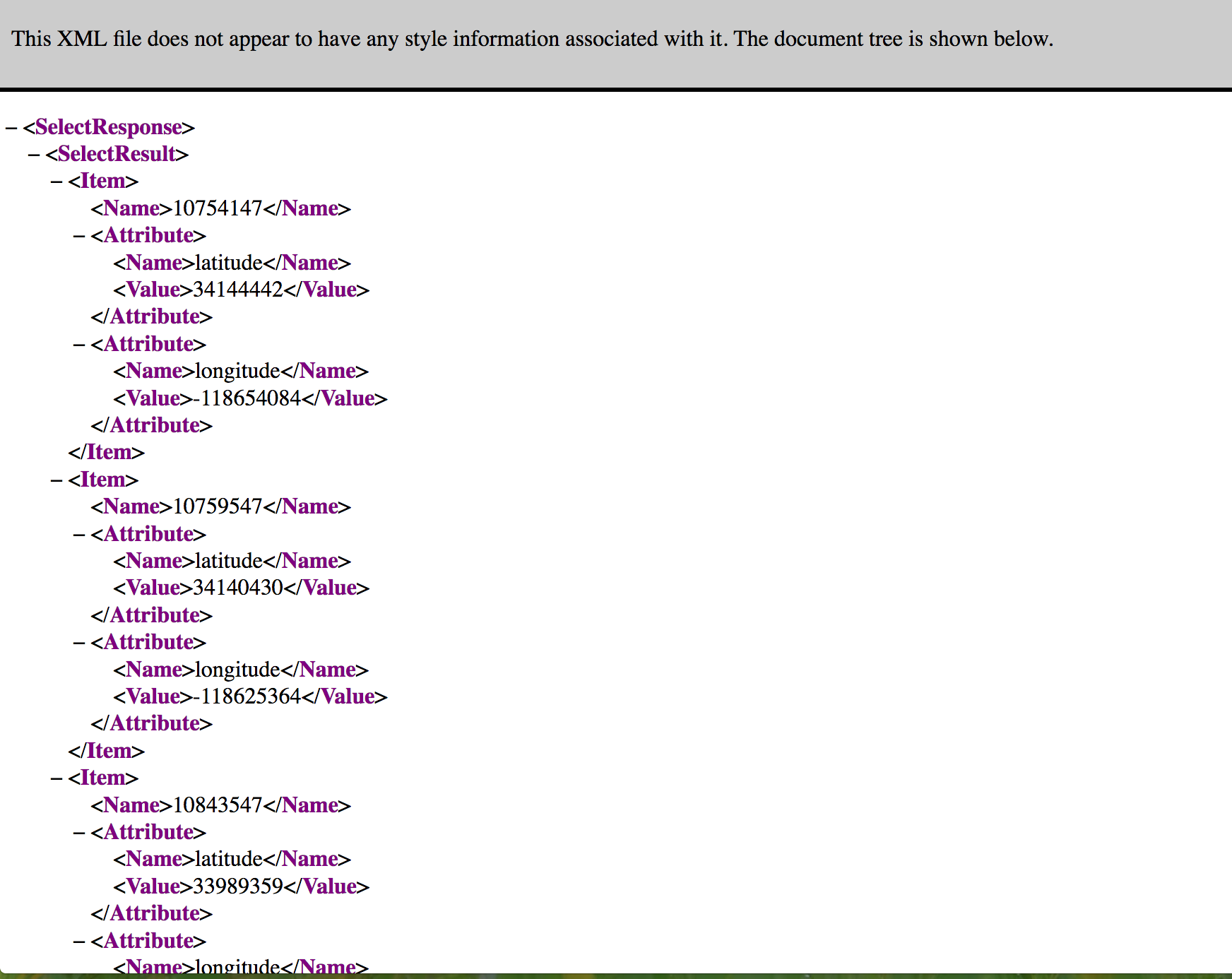
1. Configure config.json:
   1. Add AWS keys into this file
   2. Specify your AWS region
   3. Modify <cleanDataFileName>
2. sdb.py
   1. Simply run run.sh to upload your dataset into SimpleDB.



* 1. Because of the limitation of maximum 25 items per batch putting, I only uploaded 1000 items into SimpleDB.
  2. Only upload ‘parcelid’, ‘latitude’, ‘longitude’ attributes of items.

1. What’s in SimpleDB:





1. How to select from the data:
   1. Refer to SimpleDB.ipynb on GitHub
   2. Format:

select *output\_list*

from *domain\_name*

[where *expression*]

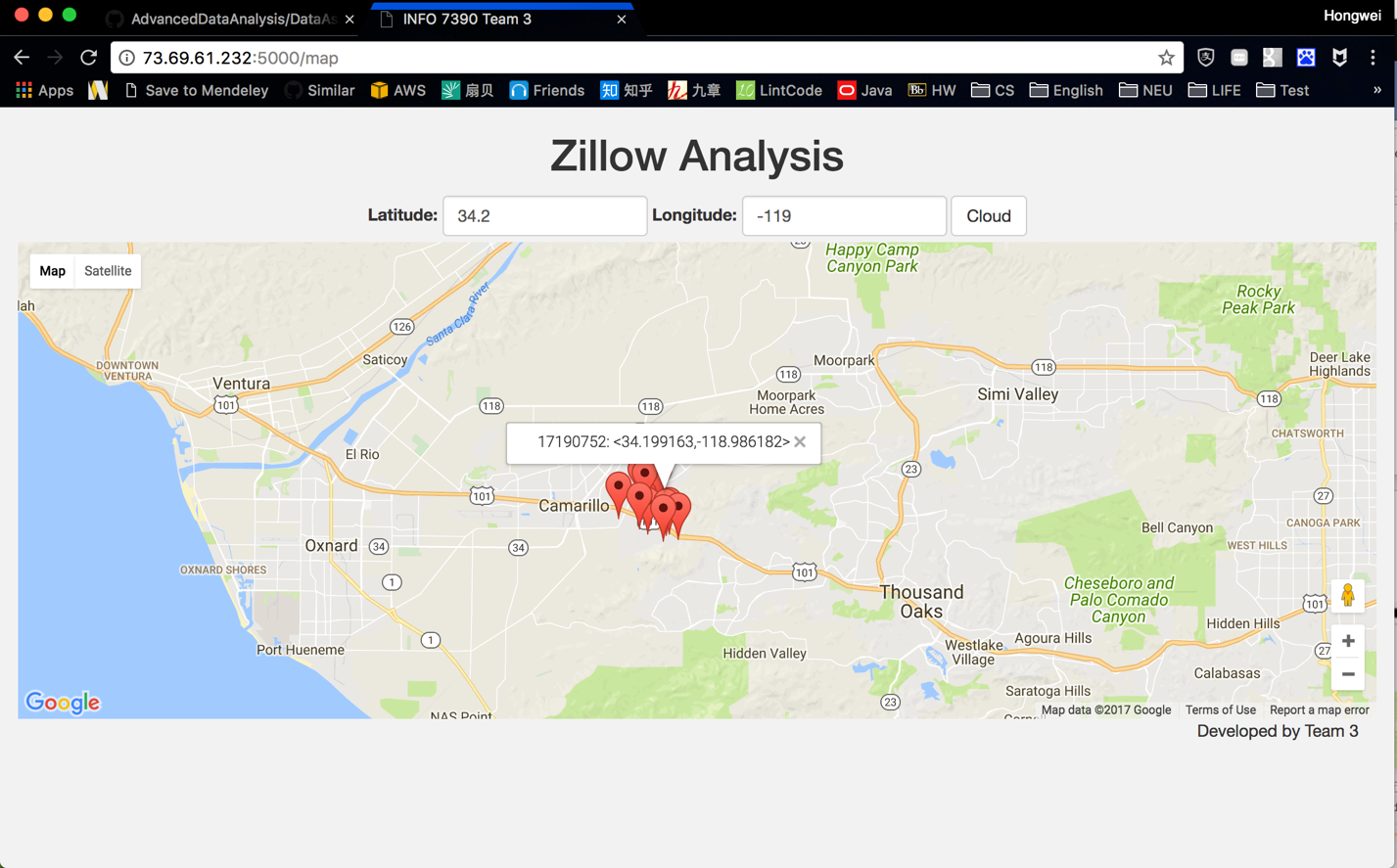
[*sort\_instructions*]

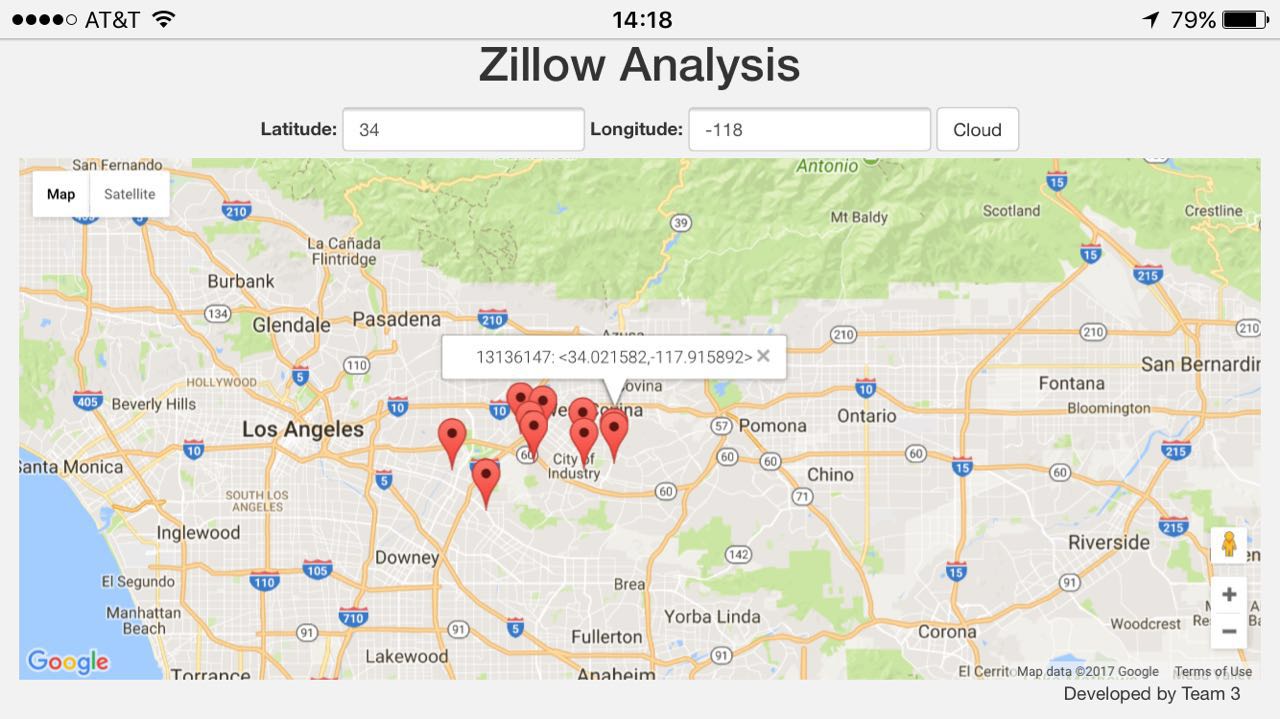
[limit *limit*]

* 1. Arithmetic Operations are not supported in SimpleDB.

**Part 3: REST API:**

1. How to realize Geospatial search:
   1. Refer to FindClosest.ipynb on GitHub
   2. Almost all the data in our dataset are located in LA
2. Integrate Google Map API:
   1. URL: 73.69.61.232:5000
      1. I deployed my service on a local machine.
      2. It won’t work after this assignment.
   2. Front end (From laptop and phone):





* 1. Back end:

