CASE 1: **Creating Data science pipelines**

Team 3 in INFO 7390

Team Members: Hongwei Hu, Guangnan Liang

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

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

Resources:

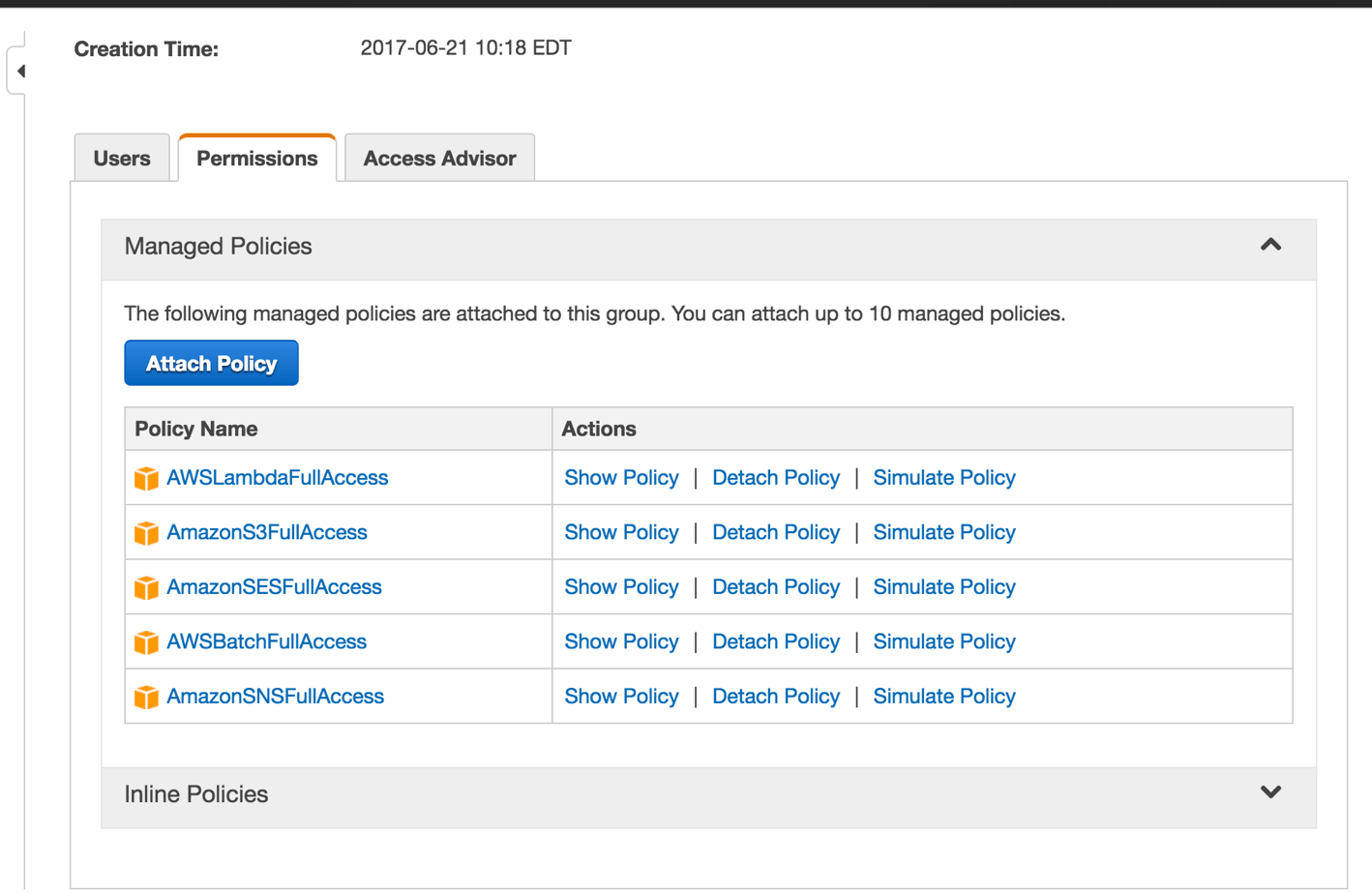
1. Our state: CT
2. CT raw data: <https://www.ncei.noaa.gov/orders/cdo/997969.csv>
3. TX raw data: <https://www.ncei.noaa.gov/orders/cdo/999575.csv>
4. Sender: [hu.hon@husky.neu.edu](mailto:hu.hon@husky.neu.edu)
5. Notification Email: [liang.g@husky.neu.edu](mailto:liang.g@husky.neu.edu)
6. CT Station ID: WBAN\_00169
7. TX Station ID: WBAN\_13910
8. CT Bucket Name: team3ctassignment1
9. TX Bucket Name: team3txassignment1

Note: Red lines means Very Important Steps

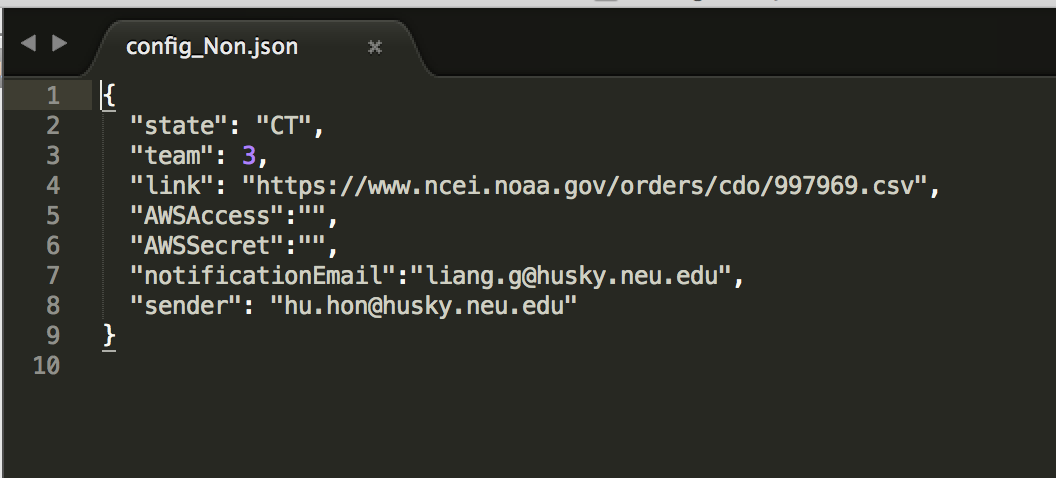
Data Ingestion & Data Wrangling:

1. Configuration (Brief)

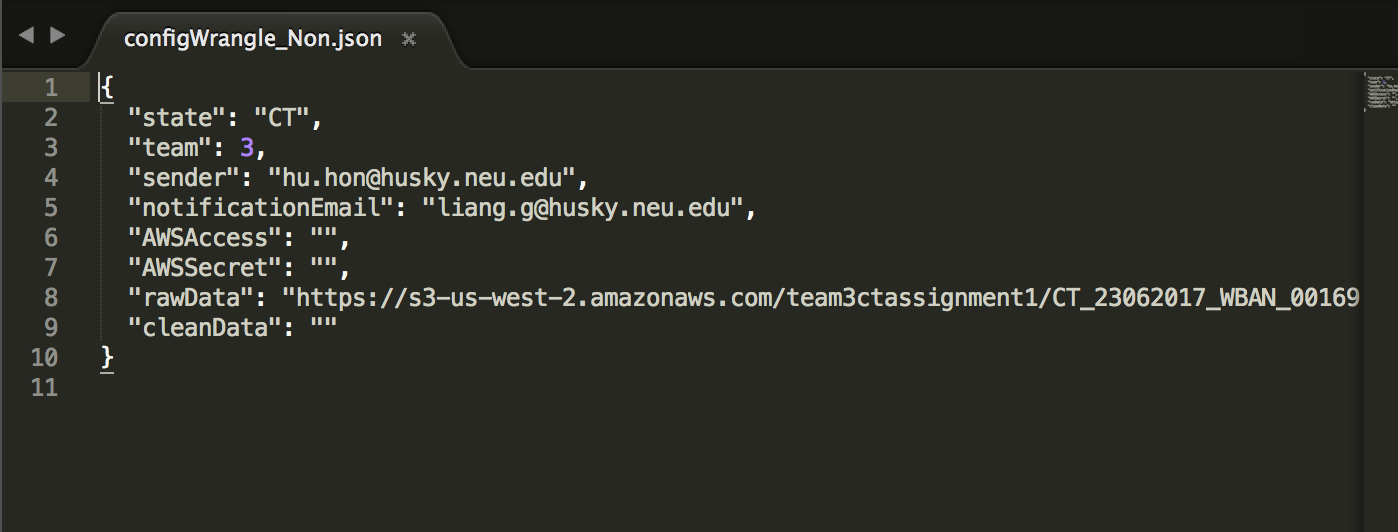
* Document: <http://boto3.readthedocs.io/en/latest/guide/migrations3.html>
* Amazon IAM Console:
* create your credentials: Add Policy



* download private key file
* Set your credentials file locally
* $ aws configure
* config.json & configWrangle.json:
* config\_Non.json & configWrangle\_Non.json
  + Add AWS Keys in this file
  + Add raw Data Link to them
  + Add Team Information
  + Add Email Sender and Receiver
  + Save them using required names
    - config.json

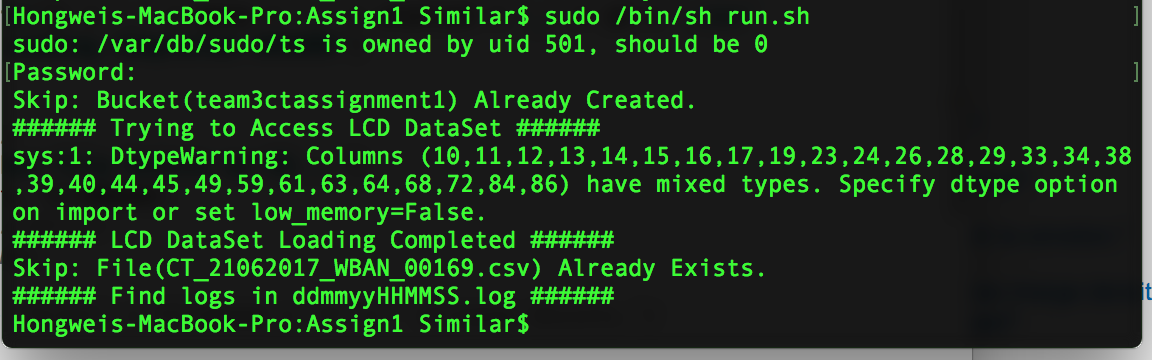


* + - configWrangle.json

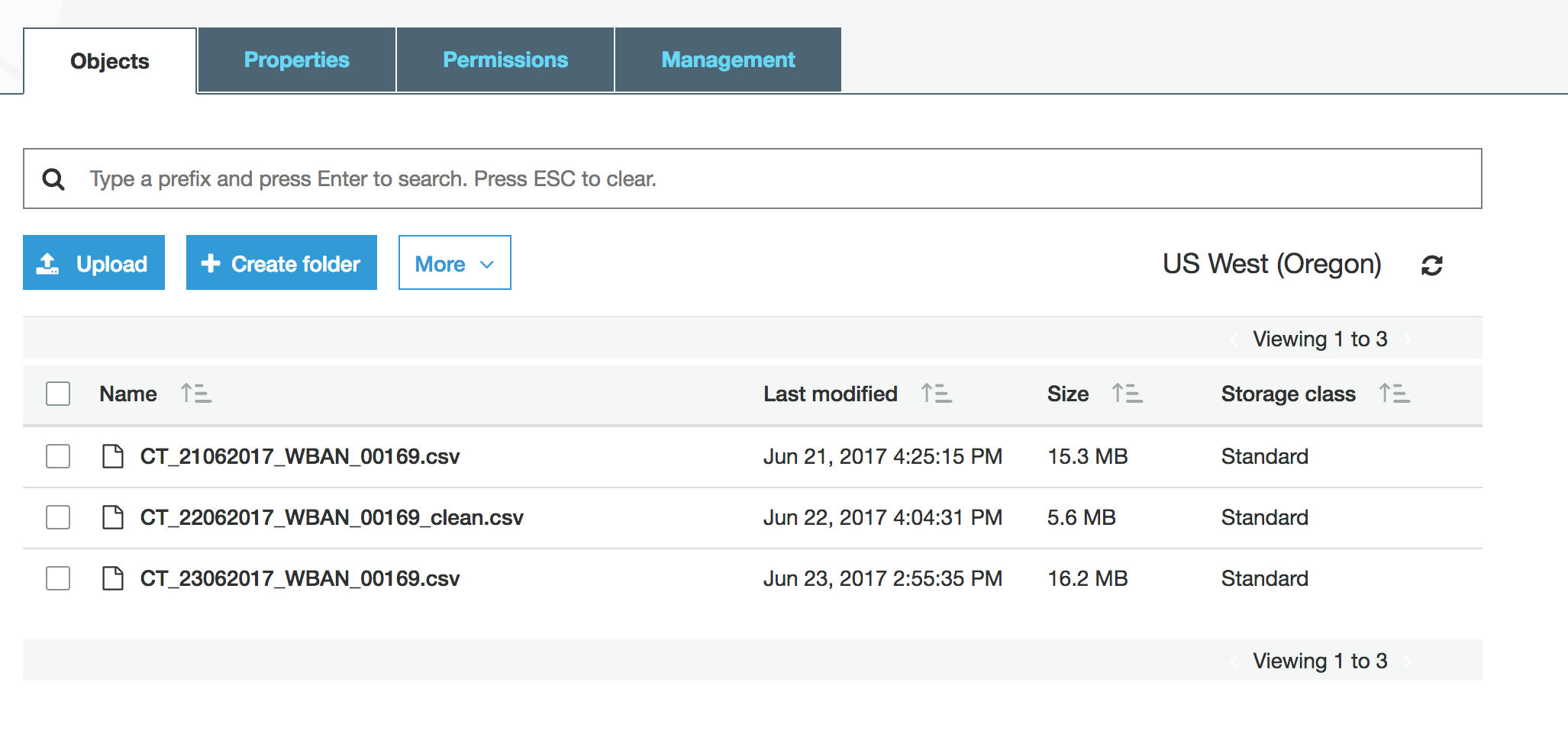


2. dataingestion.py & wrangle.py

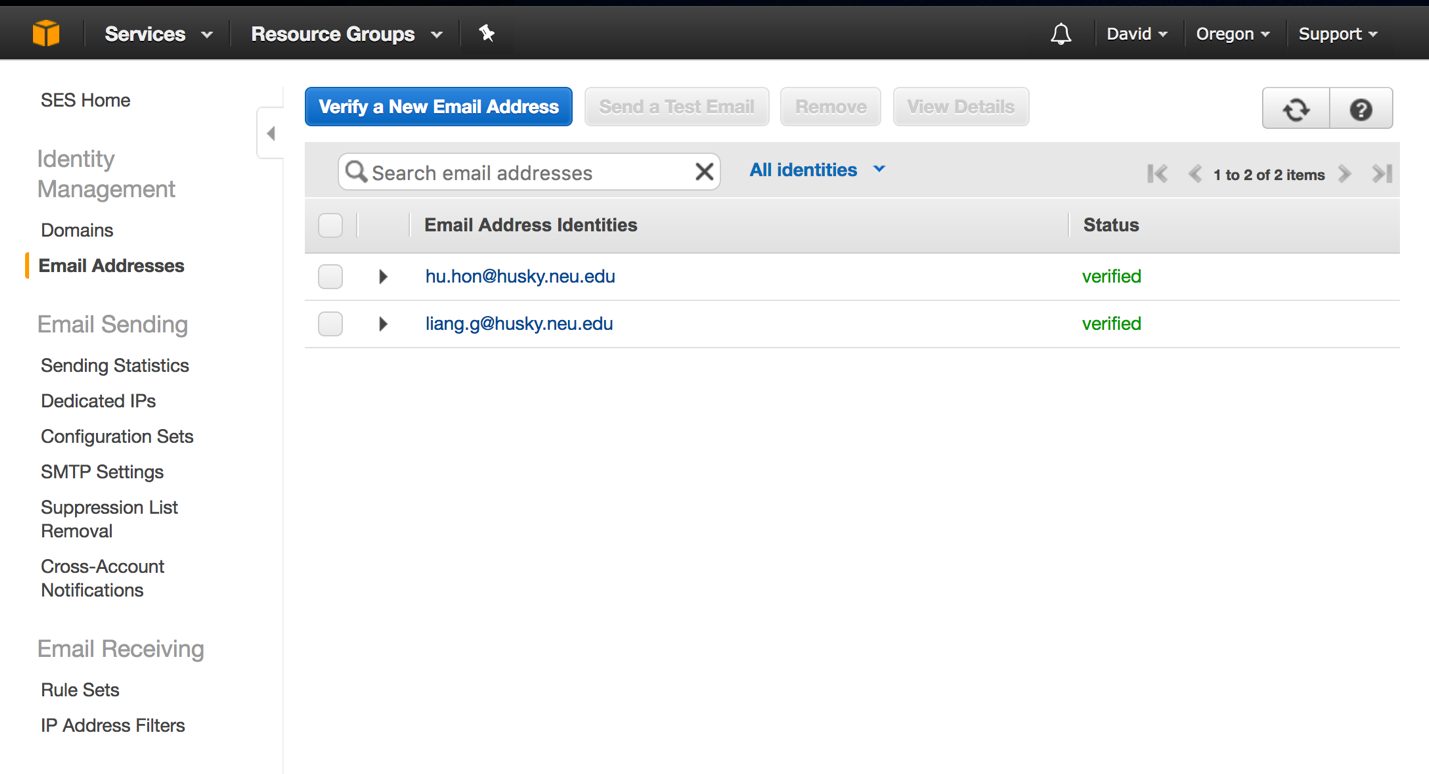
* Code: For details, please refer to code file on GitHub.
* Steps:
  1. If you have set correctly configuration files above, the Only thing you need do is:
     + $ sudo /bin/sh run.sh
     + It will do all the steps below.



* 1. Create Bucket (only in dataingestion.py): to check whether the bucket exists. If not, create a new bucket.
  2. Load raw LCD data (in both processing): to read Station Information, and use this information to check whether data of today has already existed on S3.
  3. Upload raw/clean Data to S3 (in both): If there’s no data of today on S3, upload to S3



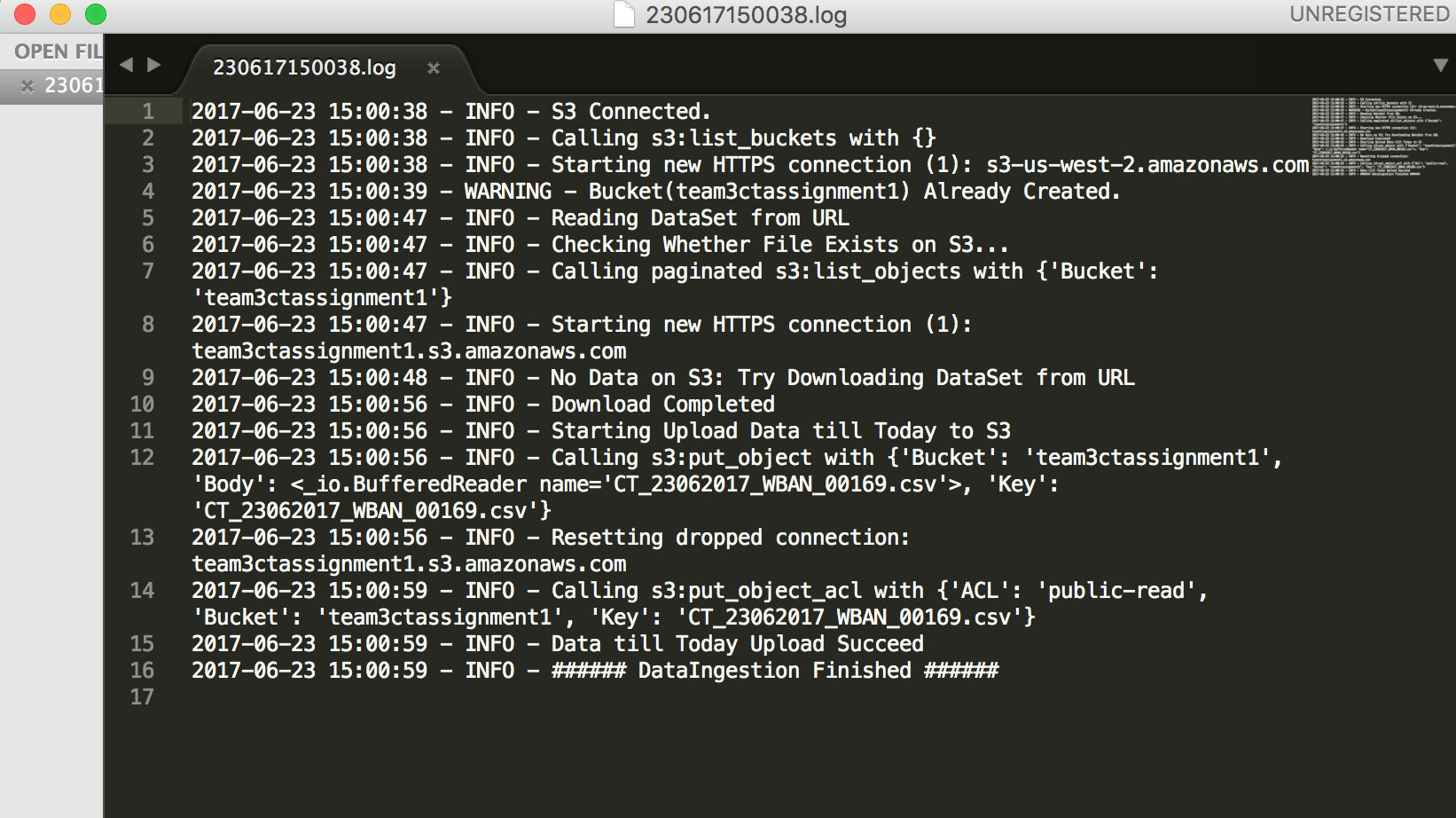
* 1. Send an Email to User (only in wrangle.py):
     + Use AWS SES for this part rather than SNS
     + You need to register your emails (both senders and receivers) in AWS SES, if not you will fail.

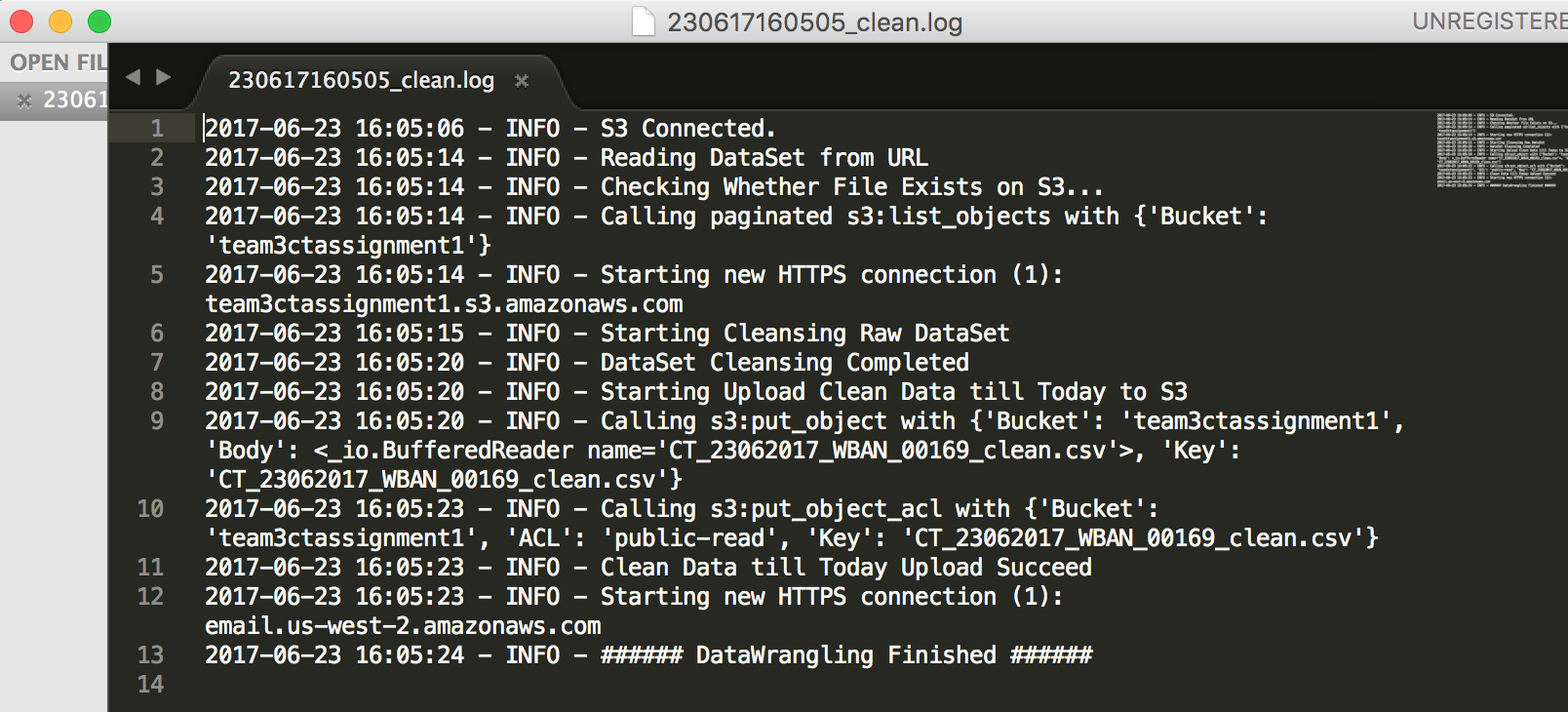


* + - In wrangle.py, don’t forget to set your AWS region to your region.
    - The email will contain the download link of clean data.



* 1. Logs into local (in both processing):





3. Docker:

1. Modify the code of Creator (Data Science in the Cloud Tool):

* in process.py:

python-pip3 --> python3-pip

RUN pip3 packages --> RUN pip3 install packages

* The code from GitHub is out of date.

2. Make Docker Images

1. Run Docker Creator:

$ cd ~/your/path/Data-Science-in-the-Cloud/webApp

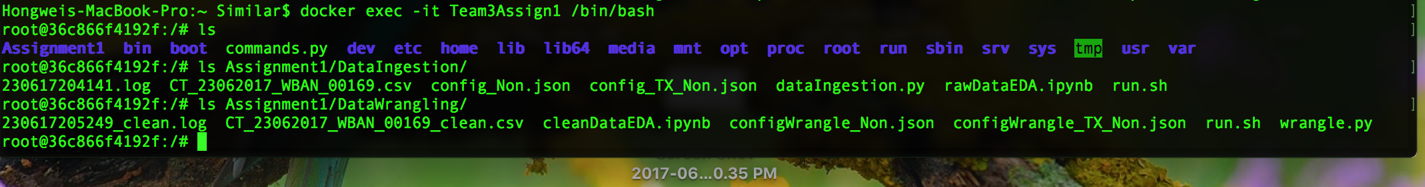
$ python manage.py runserver 8000

2. Included Packages: Python3.4

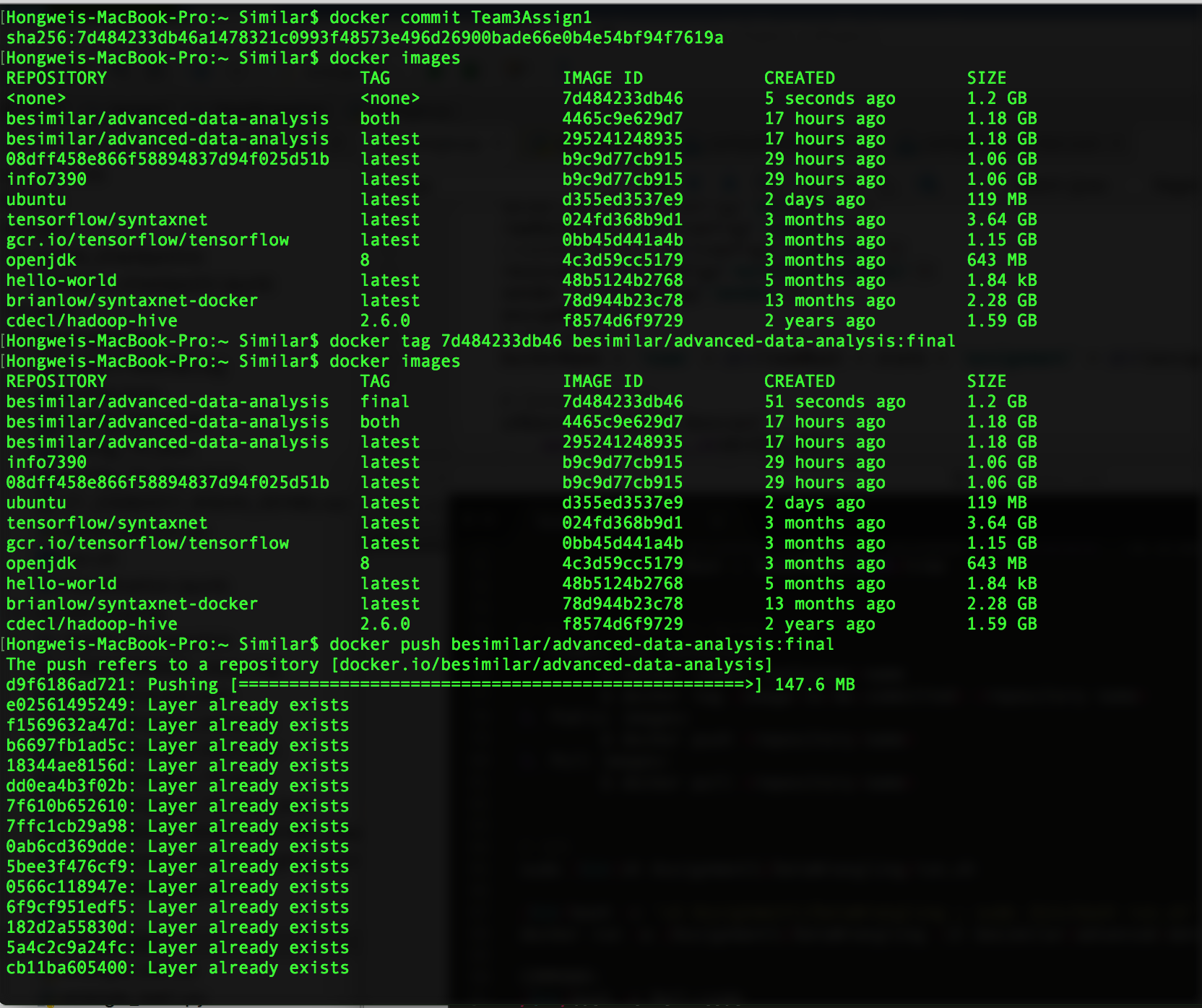
* pandas
* numpy
* Matplotlib
* IPython

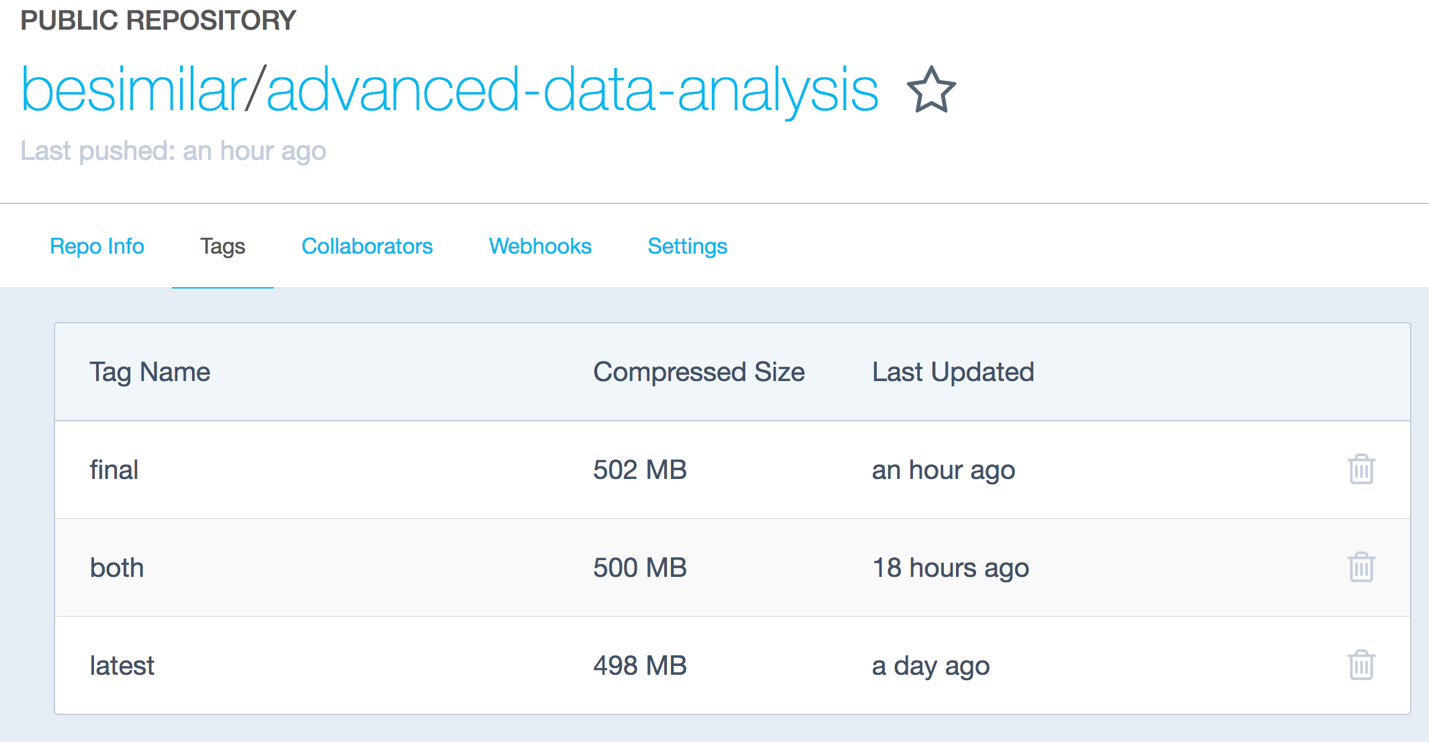
3. Run image and Configure Container:

* Run image:
  + $ docker run -it -p 8888:8888 <imageName> /bin/bash
  + port 8888 for Jupyter notebook
* Install additional packages:
  + $ pip3 install --upgrade pip
  + $ pip3 install boto3
  + $ pip3 install jupyter
  + $ pip3 install seaborn
* Copy all your files to Container:



4. Commit Container & Publish your Docker Image





4. Run Both Processing in Containers:

1. Get Images from Docker Hub:

$ docker pull besimilar/advanced-data-analysis:final

1. Run images: (set port for jupyter notebook)

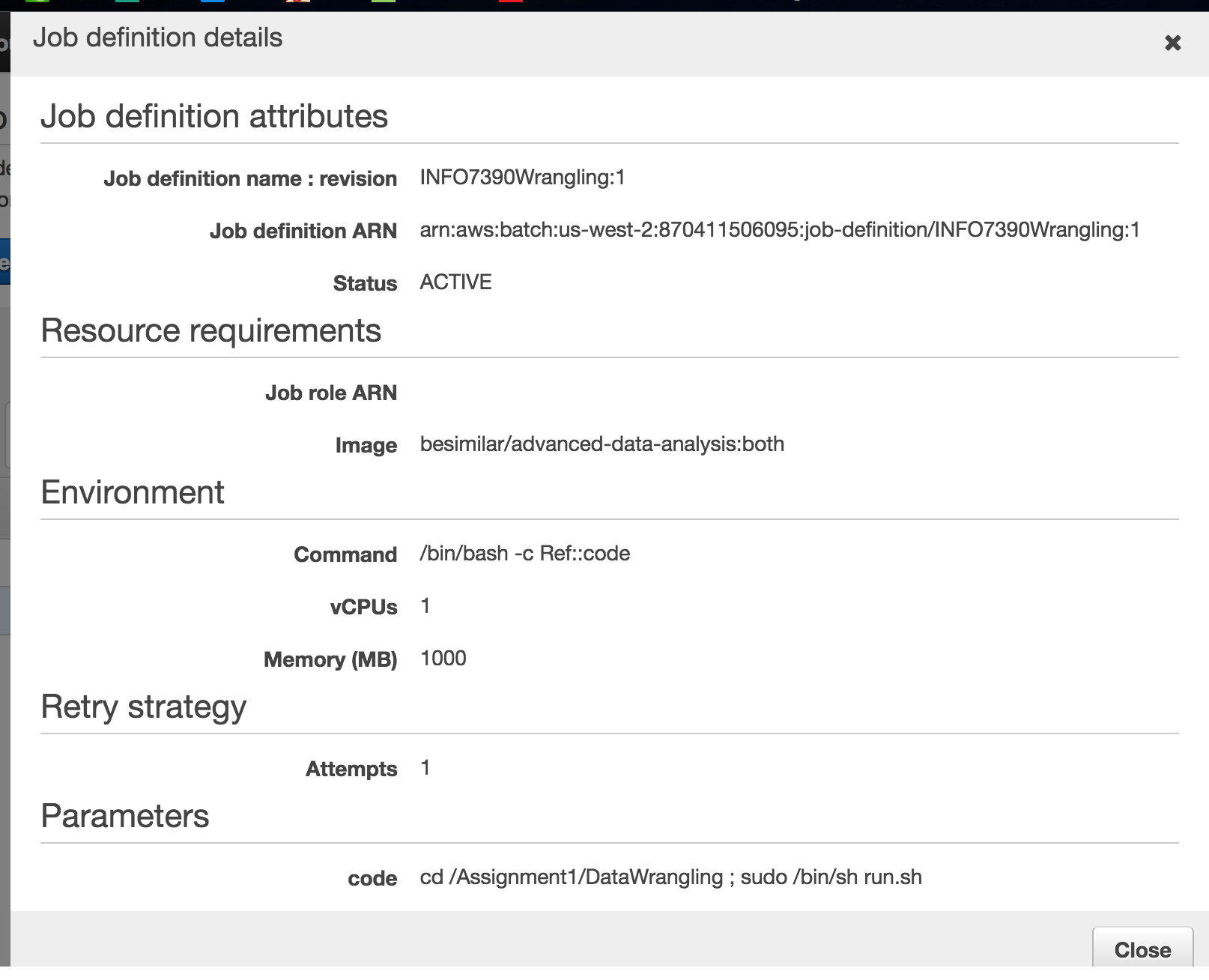
$ docker run -it -p 8888:8888 <imageName> /bin/bash

1. Run data processing: refer to Steps in (Step 2 - dataingestion.py & wrangle.py)
2. Run data analysis:

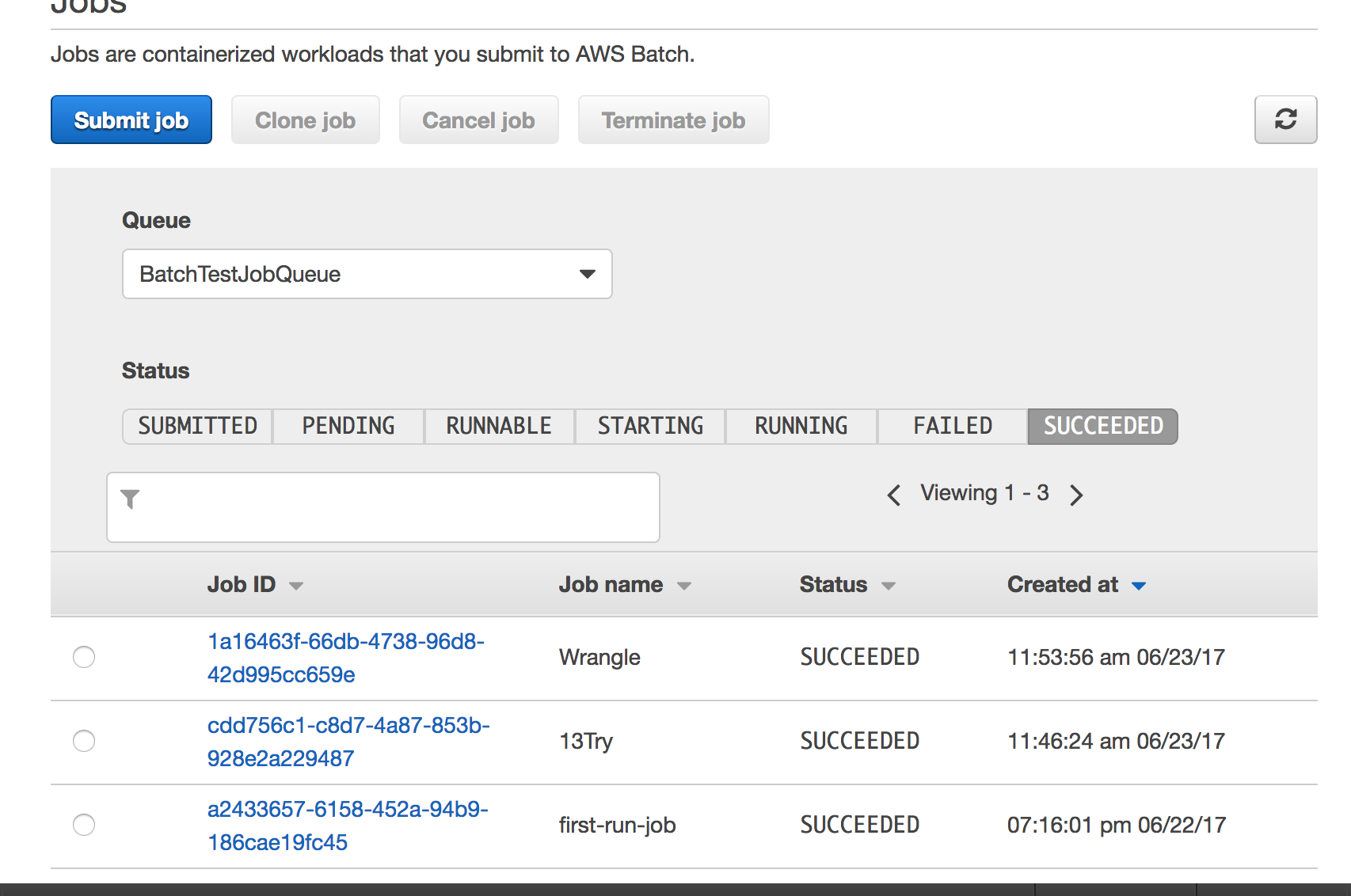
* $ jupyter notebook --ip 0.0.0.0 --no-browser --allow-root
* Summary:
  + Docker : docker run -it -p 8888:8888 image:version
  + Container : jupyter notebook --ip 0.0.0.0 --no-browser
  + Host : localhost:8888/tree‌​
* You can access notebook on your local browser using Host address.

5. AWS Batch:

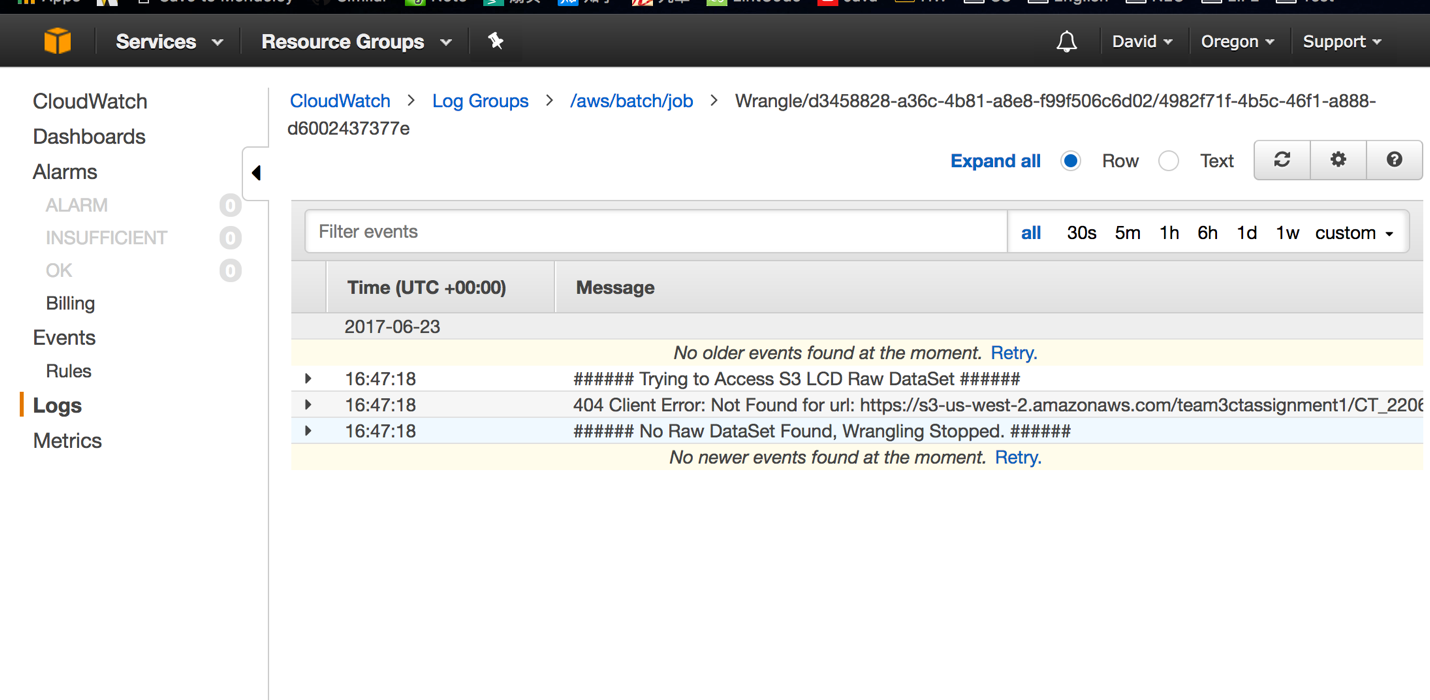
1. Job Definition:



2. Jobs:



3. Results:

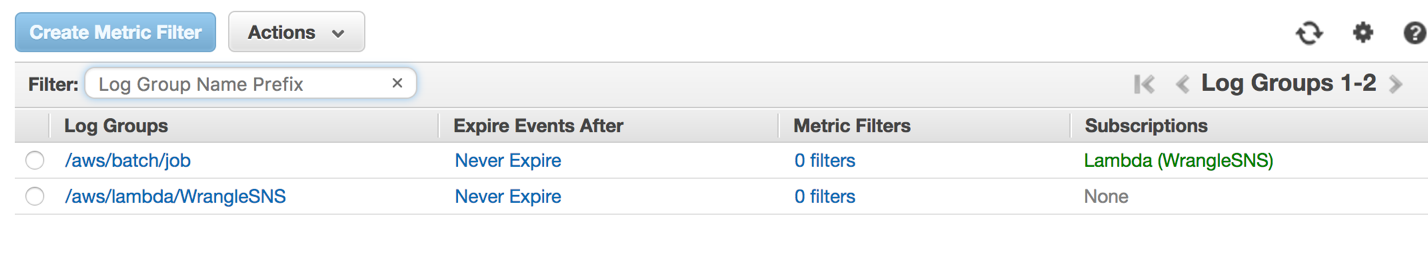


The AWS Batch job did nothing here (it stopped.), because I did not upload my AWS key in Docker images, and I did not modify the configuration File.

4. Send Email:

* I use AWS SES rather than SNS
* Details refer to Step 5 in (2. dataingestion.py & wrangle.py)

* I tried AWS Lambda: using it to subscribe jobs, but it’s too complicated.



6. Run Processing on another state: (TX of Team4):

1. Modify config\_TX\_Non.json and configWrangle\_TX\_Non.json

* Add Parameters (Links, etc)
* Save them using required names.

2. Result:

