Build and Deploy an Sentiment Analysis of Reviews [AWS and Python]

Steps:

1. Build a model on your local box (Amazon Fine Food reviews) and store the model and other key model related variables in .pkl files
2. Launch a micro instance on AWS.
3. Connect to the AWS box [ssh]
4. Move the files to an AWS EC2 instance/box [scp]
5. Install all packages needed on the AWS box.
6. Run app.py on the AWS box.
7. Check the output in the browser.

Software needed:

1. Anaconda:
    a. **Windows 64 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Windows-x86_64.exe
    b. **Windows 32 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Windows-x86.exe
    c. **Mac :** https://repo.continuum.io/archive/Anaconda3-5.2.0-MacOSX-x86_64.sh
    d. **Linux 64 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86_64.sh
    e. **Linux 32 bit:** https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86.sh
    f. **Check the previous Archives of Anaconda:** https://repo.continuum.io/archive/

2. Packages needed:

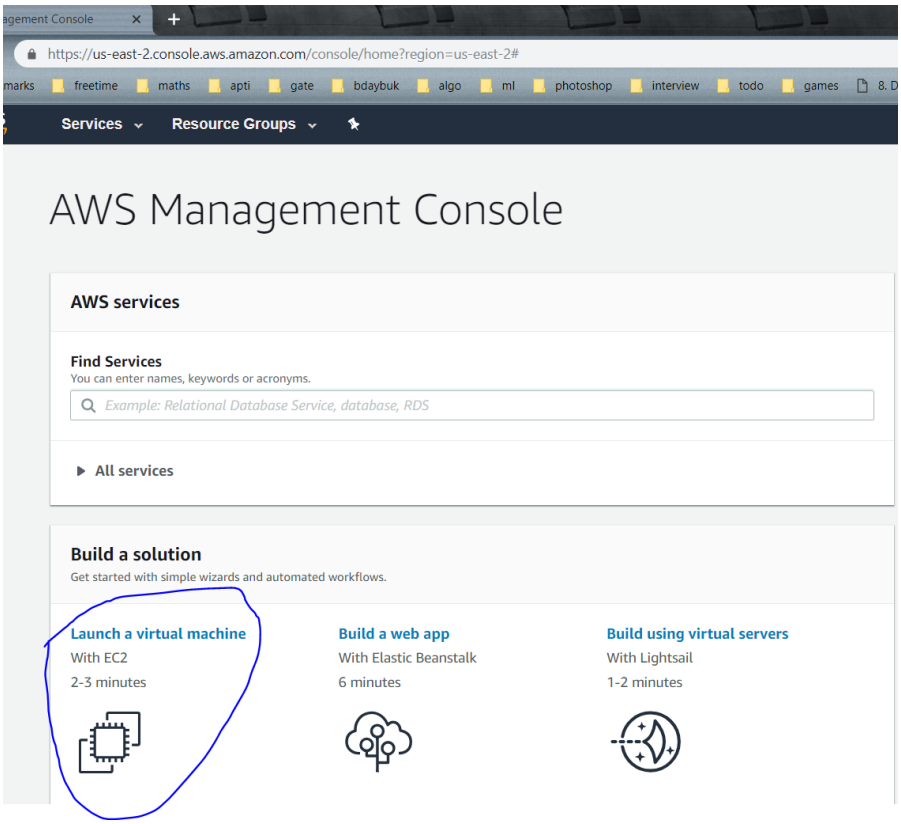
1. pip3
2. pandas
3. numpy
4. sklearn
5. beautifulsoup4
6. lxml
7. flask
8. regex

you can copy all these packages and try like this:
https://stackoverflow.com/a/15593865/4084039

[1] Launch a micro instance on AWS.

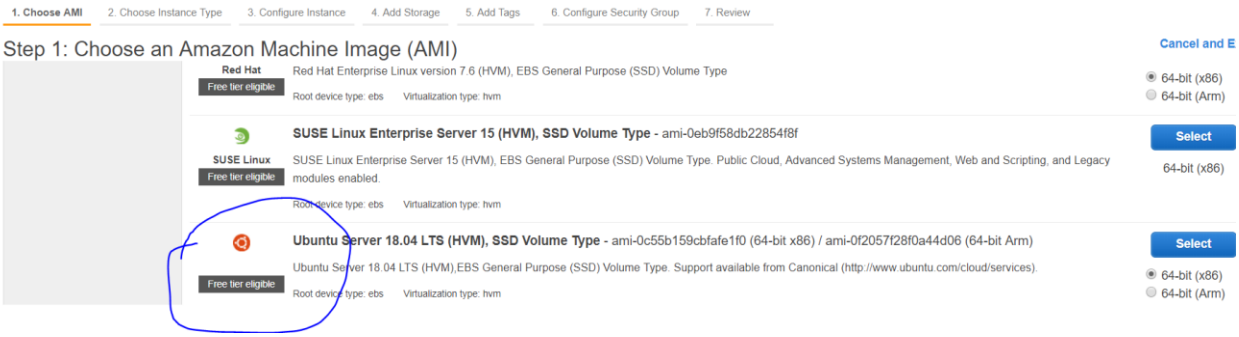
Creating an instance:

1. Create an AWS account https://aws.amazon.com, https://portal.aws.amazon.com/billing/signup#/start
2. Login: https://console.aws.amazon.com
   After login:



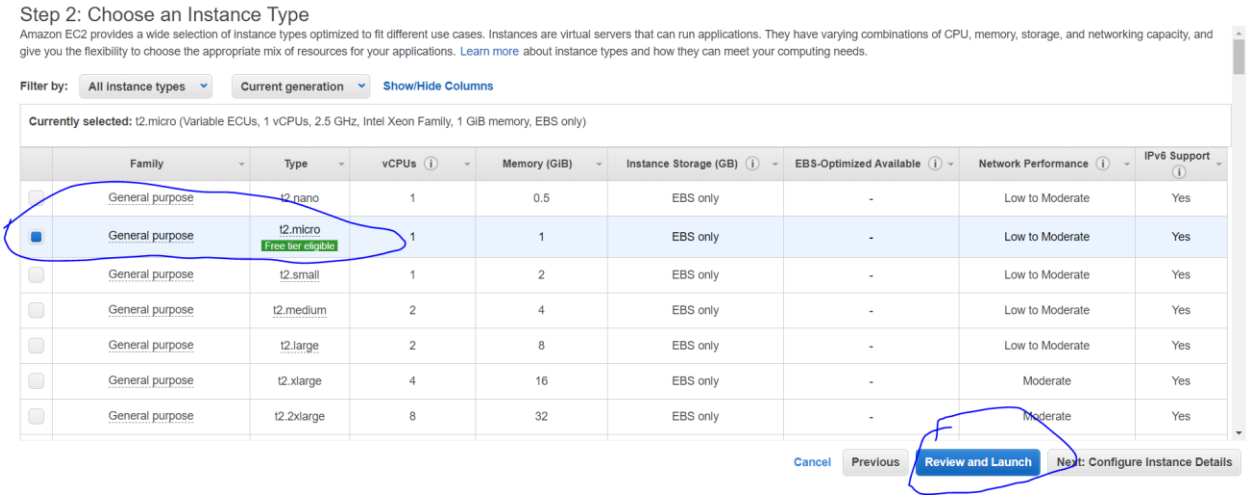
Launch the EC2 instance

3. Choose the ubuntu free tire



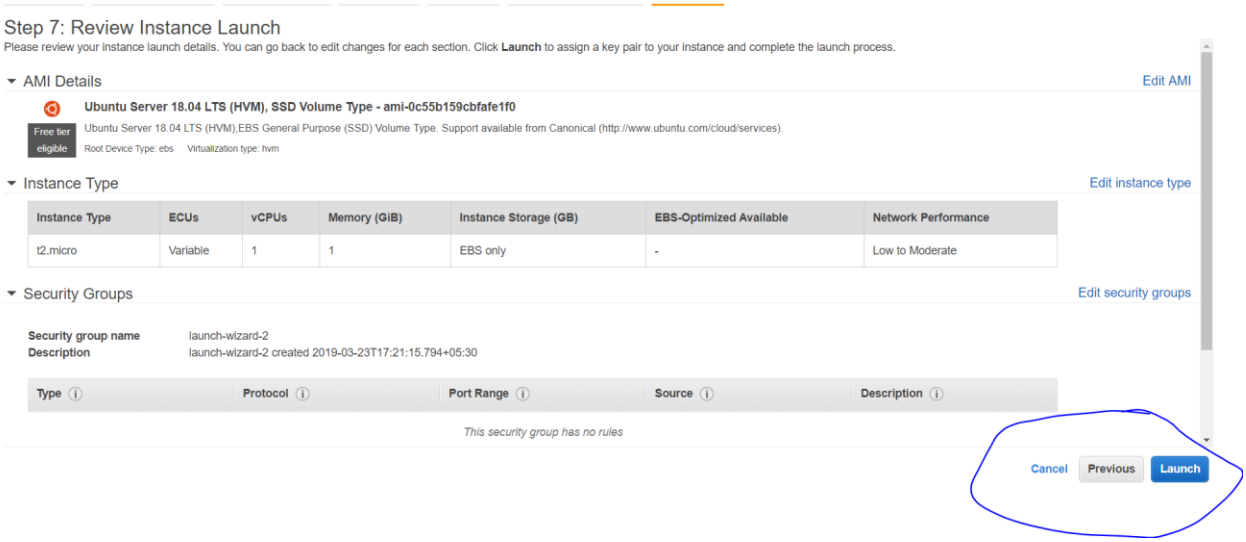
Click on select

4. Choose t2.micro free tier eligible



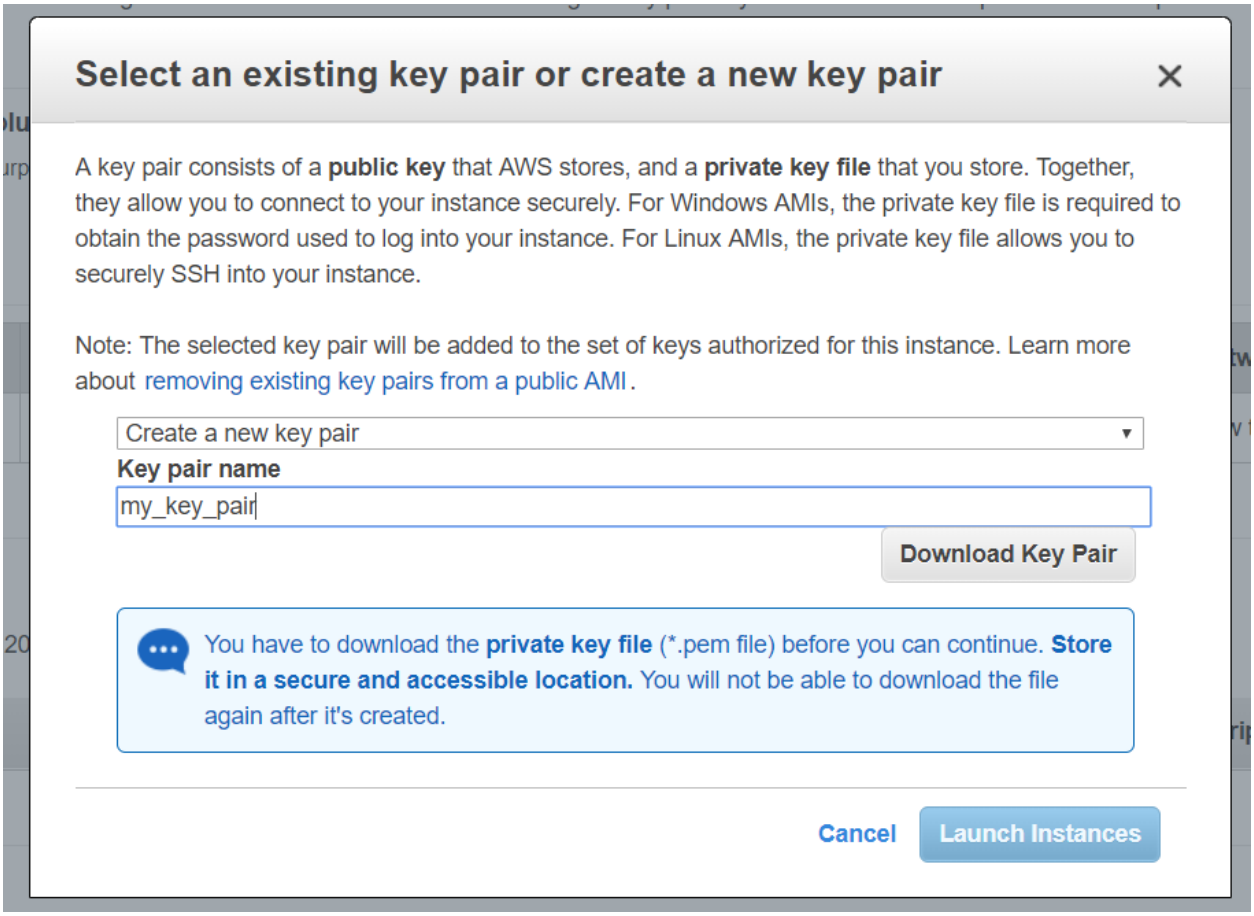
Click on review and launch

5.



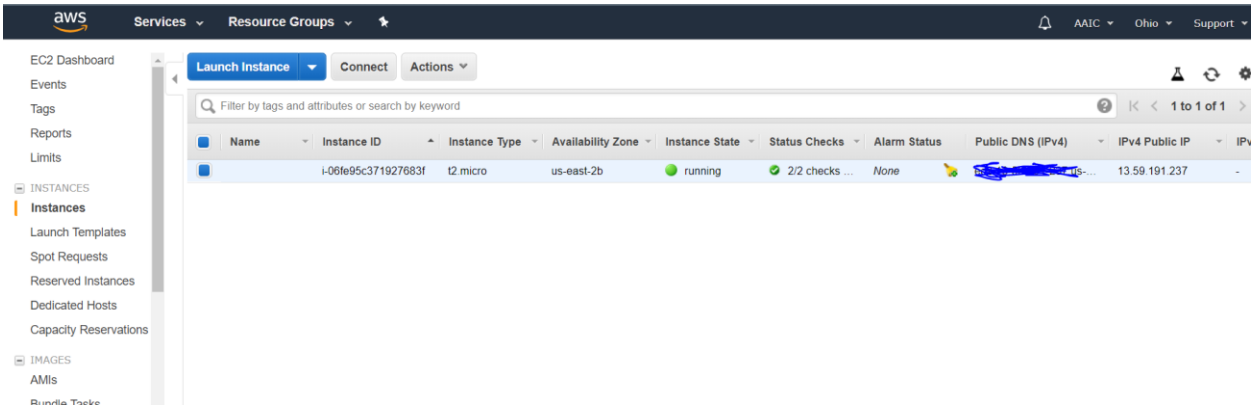
Click on launch

6.



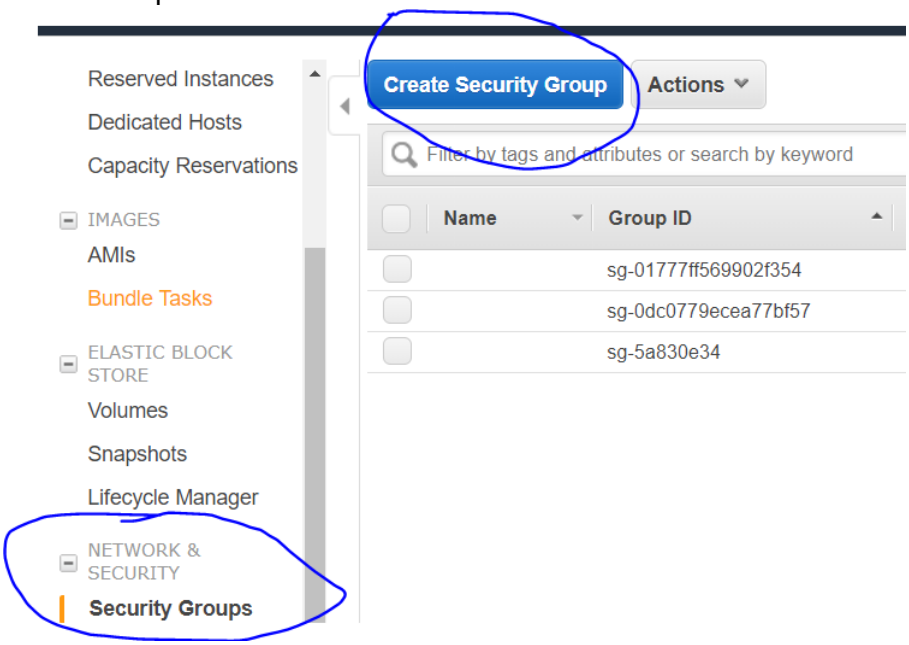
Click on “Download Key Pair” and save the .pem file then click on “Launch Instance”

7.

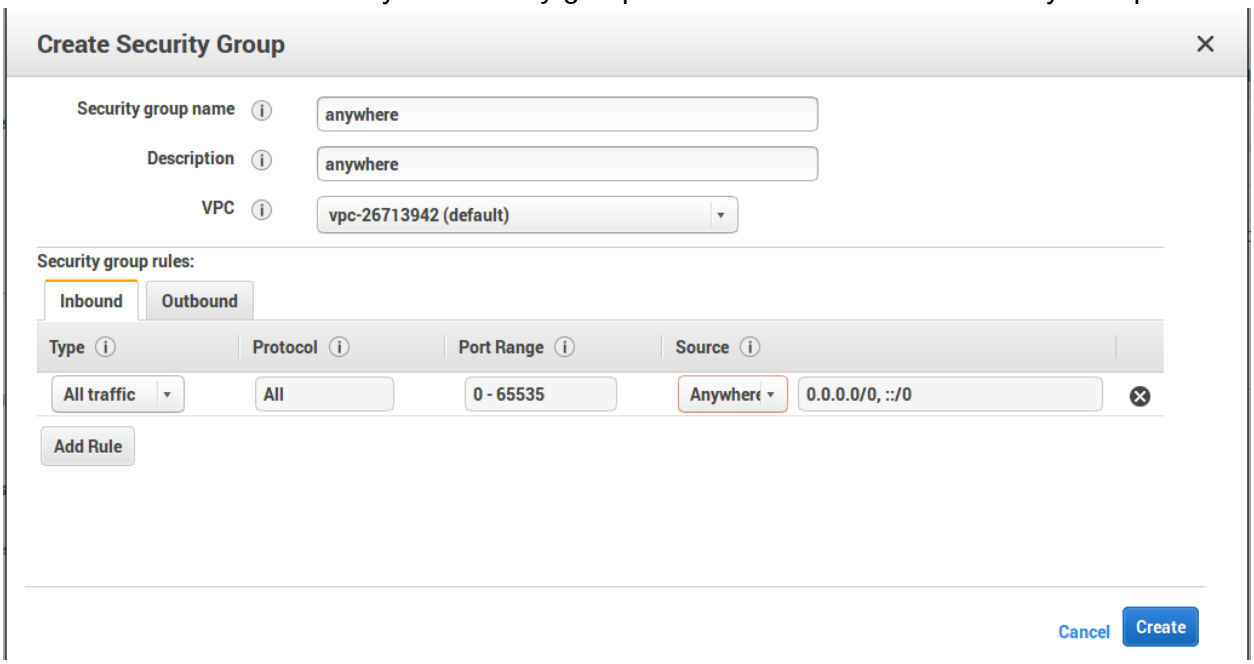


You will see this screen, you have successfully launched the an EC2 instance, now we need to launch an flask api in it

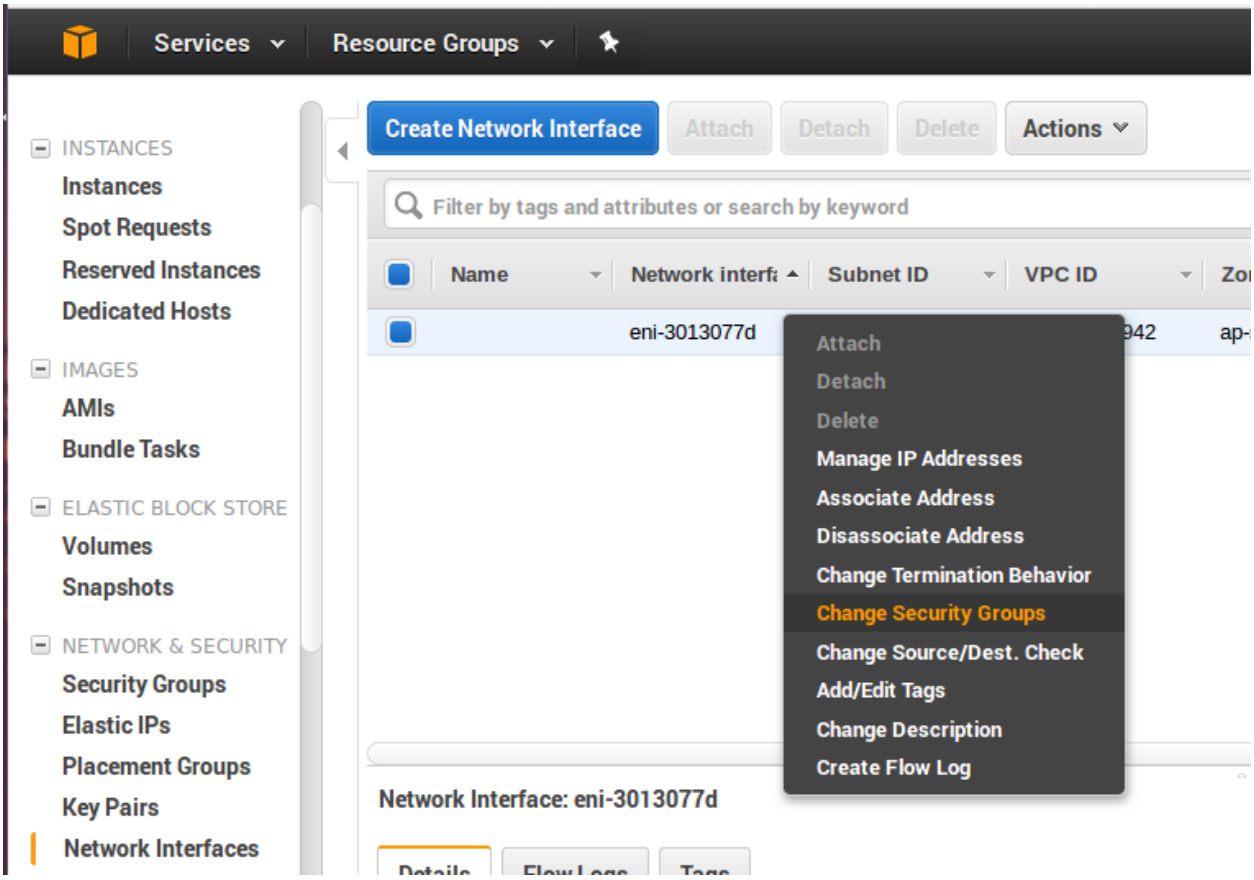
8. Final step:



Select the “Network & security” -> Security groups and then click “Create Security Group”



Then add the specific security group to **network interface**



Change Security Groups

Network Interface eni-06d51045198e42093

Security groups* sg-0dc0779ecea77bf57 sg-01777ff569902f354

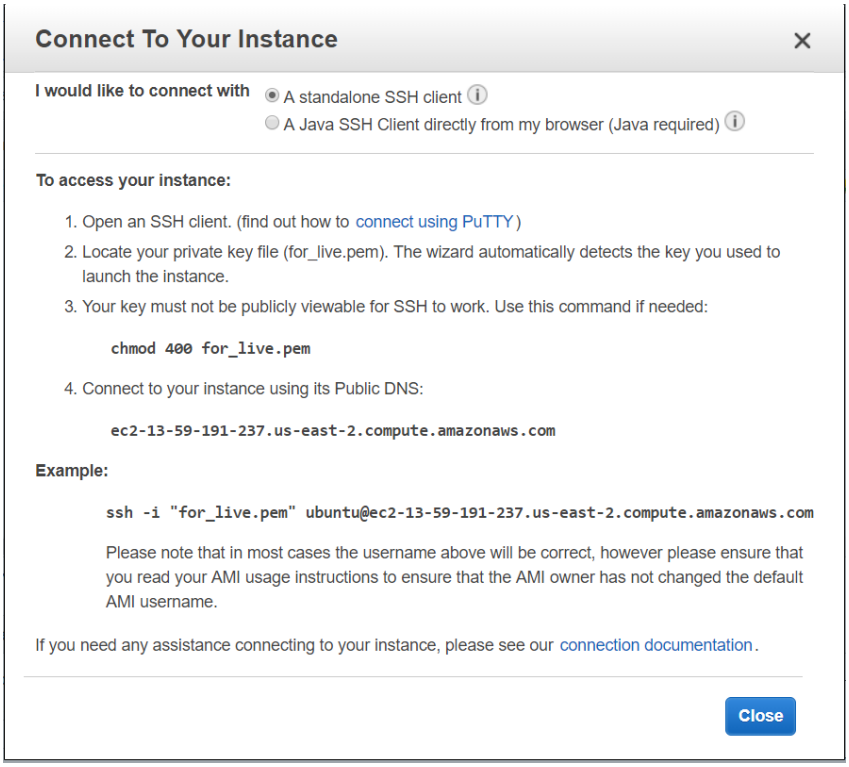
Filter by attributes or search by keyword — 1 to 3 of 3

Group ID	Group name	Description
sg-01777ff569...	anywhere	anywhere
sg-0dc0779ec...	launch-wizard-1	launch-wizard-1 created 2019-03-23T12:36:12.510+05:30
sg-5a830e34	default	default VPC security group

* Required

Cancel Save

[2] Connect to the AWS box



[3] Move the files to an AWS EC2 instance/box []

Command line to copy files

C:\Users\Asus\OneDrive\Desktop>**scp -i "for_live.pem" -r AFR http://ec2-3-14-72-94.us-east-2.compute.amazonaws.com:~/**

app.py	100%	3888	16.9KB/s	00:00
count_vect.pkl	100%	31MB	2.6MB/s	00:11
model.pkl	100%	79KB	312.4KB/s	00:00
model.py	100%	4864	19.7KB/s	00:00
index.html	100%	332	1.4KB/s	00:00

```
C:\Users\Manasa>scp -i my_key.pem -r AFR ubuntu@ec2-3-14-72-94.us-east-2.compute.amazonaws.com:~/
app.py                                                    100% 3888    16.9KB/s   00:00
count_vect.pkl                                            100%   31MB   2.6MB/s   00:11
model.pkl                                                 100%   79KB 312.4KB/s   00:00
model.py                                                  100% 4864    19.7KB/s   00:00
index.html                                                100%  332     1.4KB/s   00:00

C:\Users\Manasa>
```

[4] Install all packages needed on the AWS box.

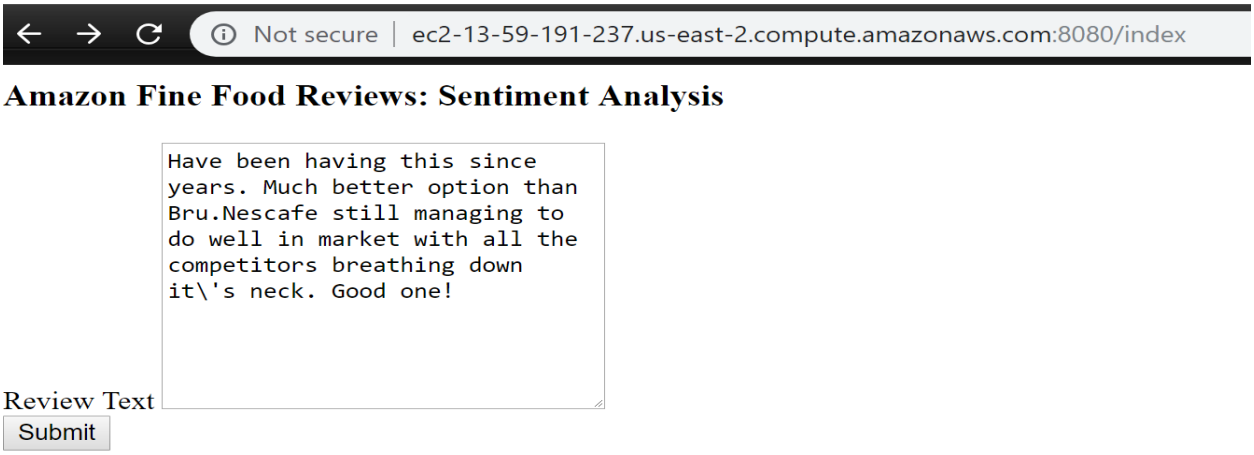
sudo apt-get install python3-pip
pip3 install <each of the following packages>
Packages needed:

- pip3
- pandas
- numpy
- sklearn
- beautifulsoup4
- lxml
- flask
- regex

[5] Run app.py on the AWS box.

```
ubuntu@ip-172-31-27-97: ~/AFR
ubuntu@ip-172-31-27-97:~/AFR$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
183.83.170.52 - - [24/Mar/2019 04:00:38] "GET /index HTTP/1.1" 200 -
```

[6] Check the output in the browser.



[7] Check the result in the browser.