TWITTER SENTIMENT ANALYSIS USING CLOUDMESH

TEAM MEMBERS:
- Gourav Shenoy
- Erika Dsouza
- Mangirish Wagle

PROBLEM DEFINITION:
Twitter is a free social networking microblogging service that allows registered members to broadcast short posts called tweets. These tweets sometimes express opinions about different topics. The aim of the project is to classify cuisine tweets from all over the world (specifically Indian, Chinese, Italian and Mediterranean) into positive, negative and neutral using cloudmesh services.

GOALS:
- Use cloudmesh services to spawn a virtual machine
- Configure hadoop HDFS to store twitter data
- Feed in live twitter data using twitter apis into hadoop cluster
- Classify twitter feeds into positive, negative and neutral.
- Perform data manipulation and create a portal with graph charts

SOFTWARE USED:
- Futuresystems Cloudmesh platform
- Python 2.7
- Hadoop HDFS
- hadoopy python library for HDFS
- Indicoio online classifier APIs
- Bootstrap.js
- Google Charts

STEPS:

1. **Spawn a virtual machine using cloudmesh service:**

   We have been contributing to the cloudmesh project and considered using some of the services that we have developed like vm management. To install clousmesh on your laptop, simply follow the guide as described in the link: http://cloudmesh.github.io/client/

   Once cloudmesh is installed, spawning a virtual machine is very trivial. Consider creating a vm on india cloud(kilo), use command:

   Create a vm on kilo cloud (since the default cloud is kilo)
   ```
   $ cm vm boot --name=<vm_name>
   ```

   Refresh the local database to get latest vm information
   ```
   $ cm vm refresh
   ```

   See the vm details
   ```
   $ cm vm list <vm_name>
   ```
   Add a floating ip to access it from the internet

```
$ cm vm floating_ip_assign <vm_name>
```

Now, once the vm shows as ACTIVE, login directly using "`ssh <user>@<floating_ip>`"

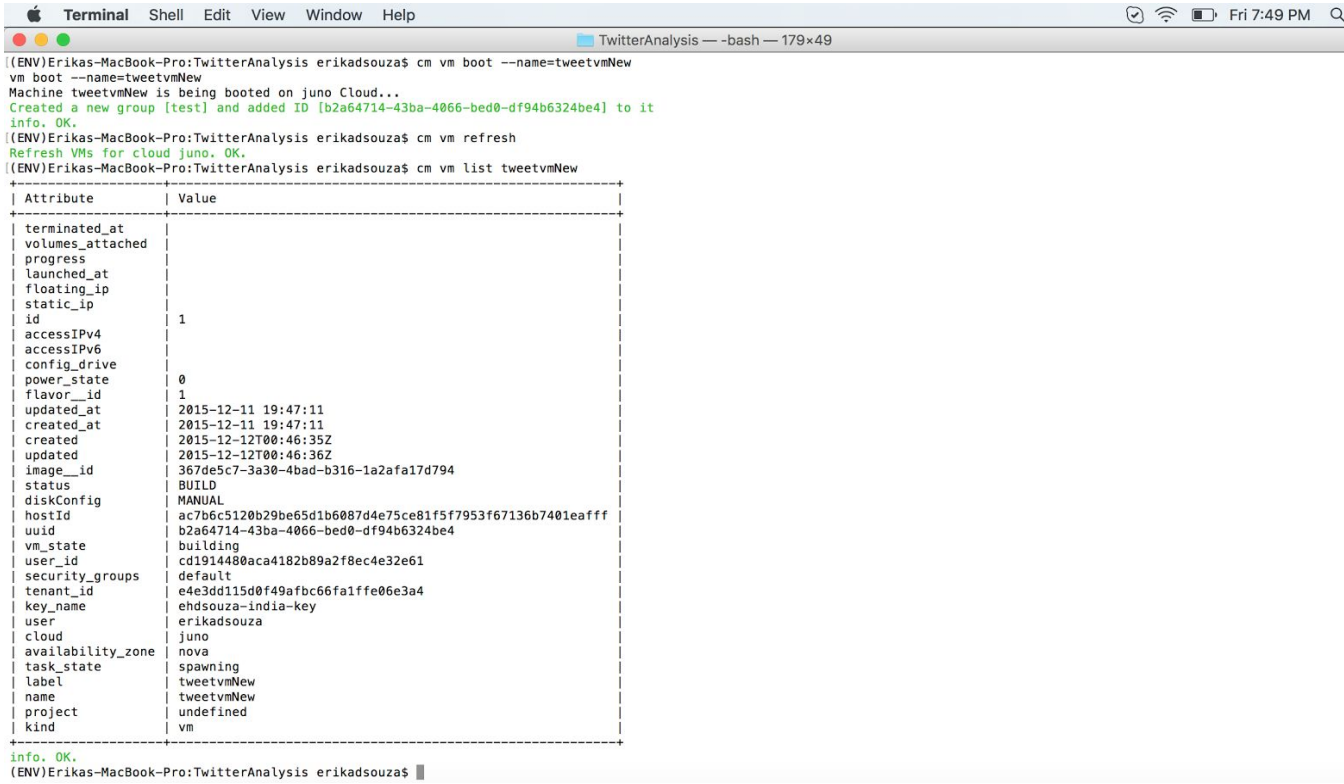


Fig: Shell commands showing creation of vm

2. **<u>Configure Hadoop HDFS to store twitter data</u>**

   Apache Hadoop is an open source framework for distributed storage and processing of large sets of data on commodity hardware. Hadoop enables businesses to quickly gain insight from massive amounts of structured and unstructured data.

   We configured the hadoop cluster following the single-node installation guide using the link: https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-on-ubuntu-13-10

   Once the installation is complete, verify that all the nodes are up:

Fig: jps command showing hadoop nodes

We used a python wrapper called Hadoopy for hadoop streaming of data. Install hadoopy using: sudo pip install -e git+https://github.com/bwhite/hadoopy#egg=hadoopy

3. Collect live twitter data using streaming apis:

a) Getting Twitter API Keys:

In order to access Twitter Streaming API, we need to get 4 pieces of information from Twitter: API key, API secret, Access token and Access token secret. Follow the steps below to get all 4 elements:

- Create a twitter account if you do not already have one.
- Go to https://apps.twitter.com/ and log in with your twitter credentials.
- Click "Create New App"
- Fill out the form, agree to the terms, and click "Create your Twitter application"
- In the next page, click on "API keys" tab, and copy your "API key" and "API secret".
- Scroll down and click "Create my access token", and copy your "Access token" and "Access token secret"

b) Connecting to twitter streaming API and downloading data:

Below is a code snippet showing how a python library called Tweepy is used:

```
class TwitterFeeds(object):
   @classmethod
   def get_auth(cls):
       auth = OAuthHandler(consumer_key, consumer_secret)
       auth.set_access_token(access_token, access_token_secret)
       return auth

   @classmethod
   def get_tweets(cls, keyword):

       global tweet_file

       auth = cls.get_auth()
       listener = StdOutListener()        # define the listener
       stream = Stream(auth, listener)      # define stream object
       api = tweepy.API(auth)# define the api object

       current_milli_time = str(int(round(time.time() * 1000)))
       tweet_file = open(keyword+'_'+current_milli_time+'.txt', 'a')

       try:
           # get past tweets, max 1million
           result = tweepy.Cursor(api.search, q=keyword).items(1000000)
           for tweet in result:
               tweet_file.write(tweet.text.encode("UTF-8"))
               tweet_file.write('\n')

           # run live feeds
           stream.filter(track=[keyword])
       except Exception as ex:
           print(ex.message, ex)
       finally:
           tweet_file.close()
```

Points to Note:-

1. The tweets are obtained in json format and stored on HDFS.
2. The twitter past feeds was fetched using twitter APIs with a usage limit of 15 calls per every 15 mins.

4. Clean data and perform sentiment analysis

The data obtained from the twitter api contained a lot of unstructured data, we created python scripts to process the data and get it into a structured form. The past feeds and the live feed tweets were then analyzed for sentiments using an online classifier APIs provided by 'Indico' (https://indico.io/docs#sentiment). We used up the free edition which provided 50000 API calls per month to analyze all the feeds for sentiments (positive, negative, neutral).

Creating an account on Indico provides you with a API key. Sample sentiment analyzer code snippet:-

```
import indicoio
indicoio.config.api_key = '43c6c90b11e05e76a9432exxxxxxxxxx'

# single example
indicoio.sentiment("indico is so easy to use!")

# batch example
indicoio.sentiment([
    "indico is so easy to use!",
    "everything is awesome!"
])
```

Sentiment Classifier:

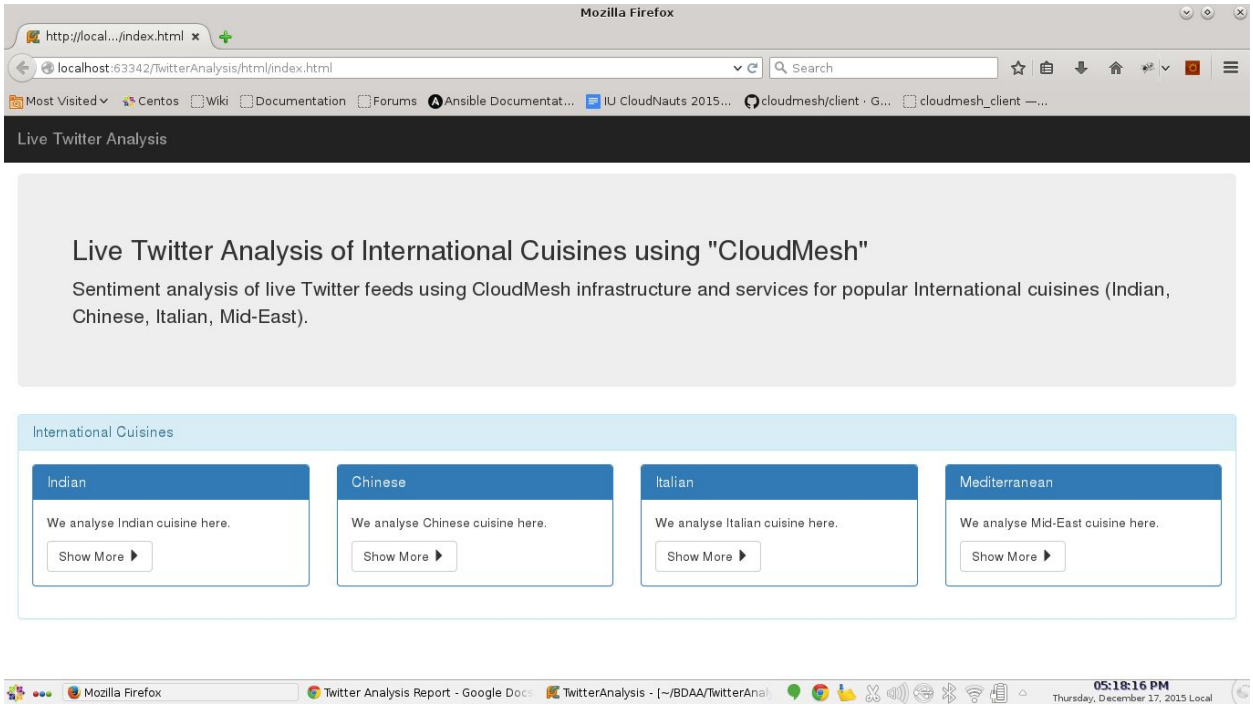
In the project, we have performed sentiment analysis of cuisines from twitter feeds and classified as positive, negative and neutral using the indico api. Below, is an example of tweet sentiment classification:

TWEET	CLASSIFICATION
"The best Cheese naan I had in my life :) #dinner"	POSITIVE
"I hated the dosa at Haandi #yucks#notagain"	NEGATIVE
"Having biryani for lunch #biryani#lunch"	NEUTRAL

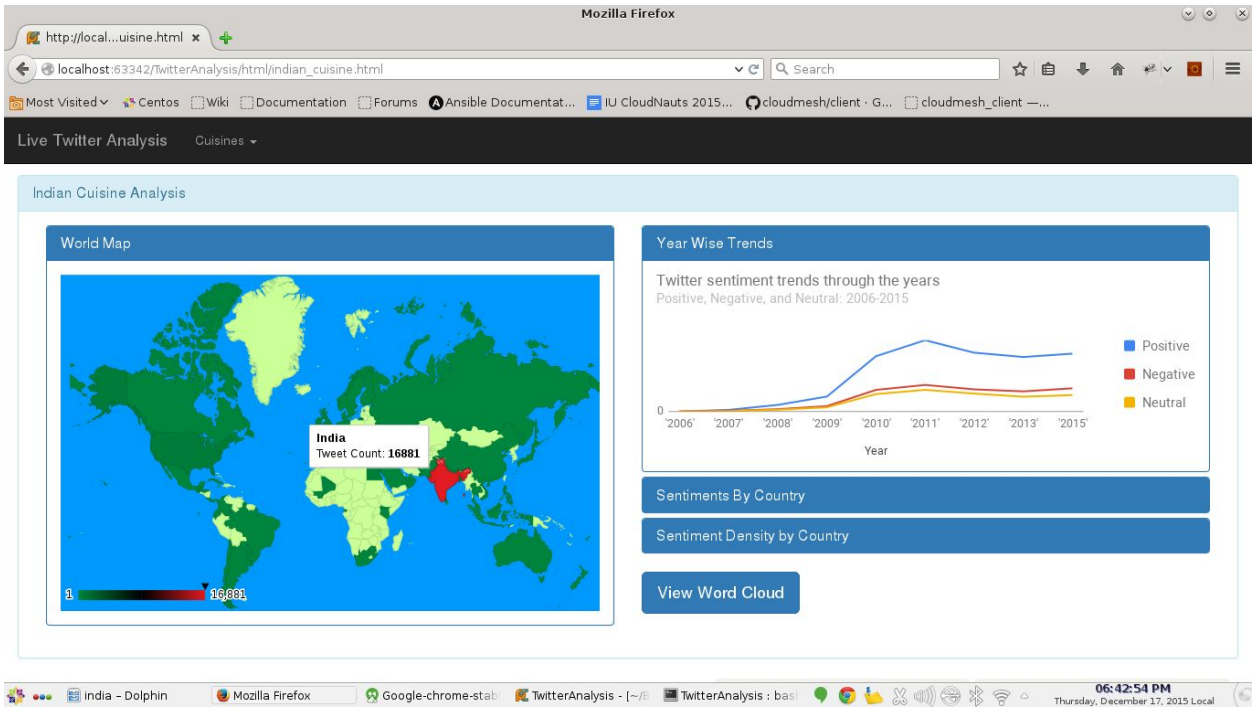
5. Results

We have put together a web portal developed using bootstrap.js and Google Charts to represent the various cuisine data that we have processed. Following are some of the snapshots of the portal.

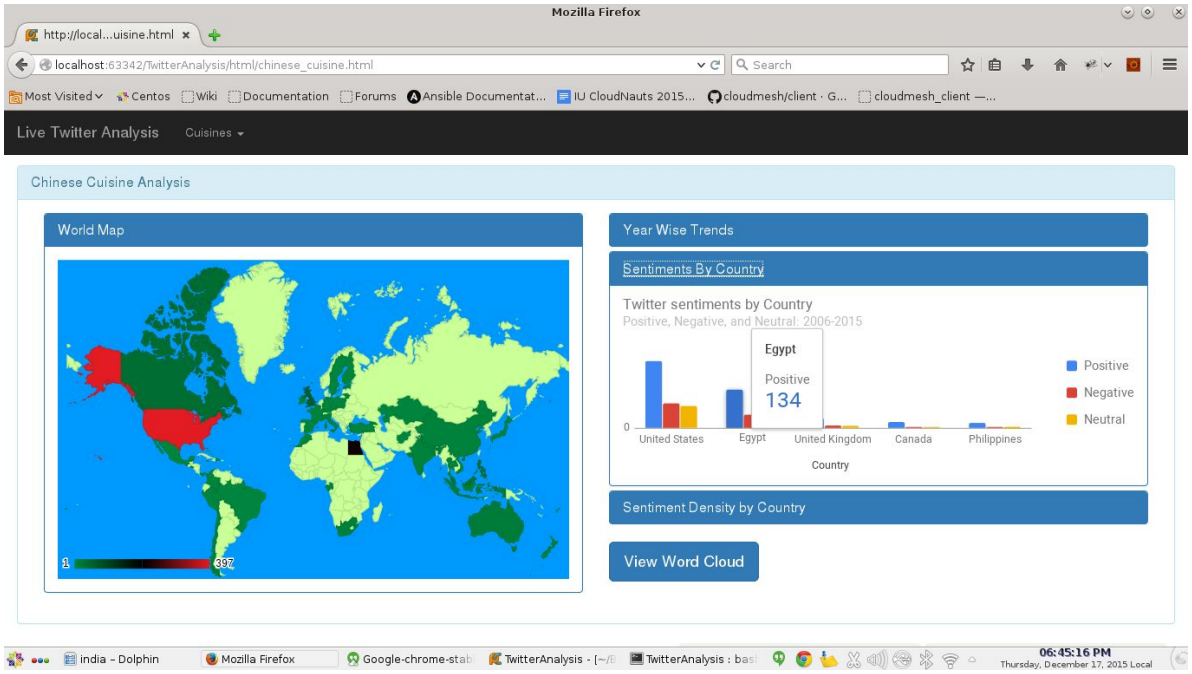
Home/ Index Page



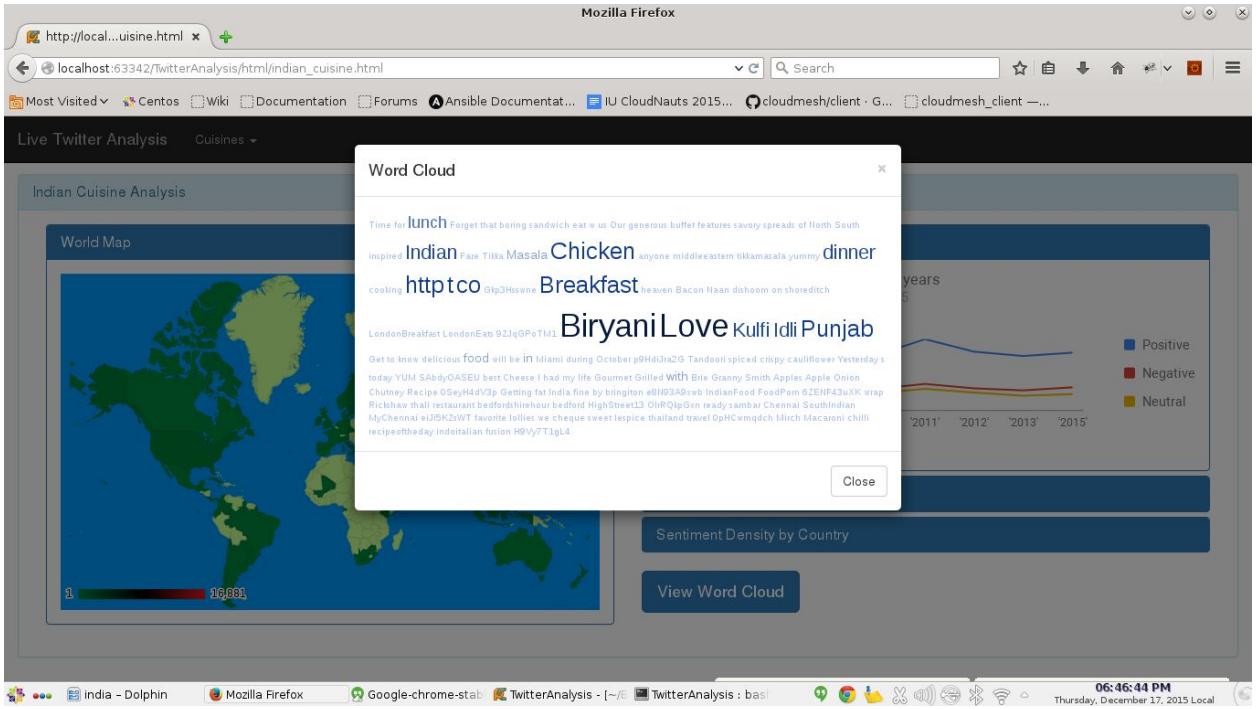
World Map Graph showing tweets from various countries using color patterns:-



Bar Chart showing Positive, Negative, Neutral tweets from countries with top 5 number of tweets:-

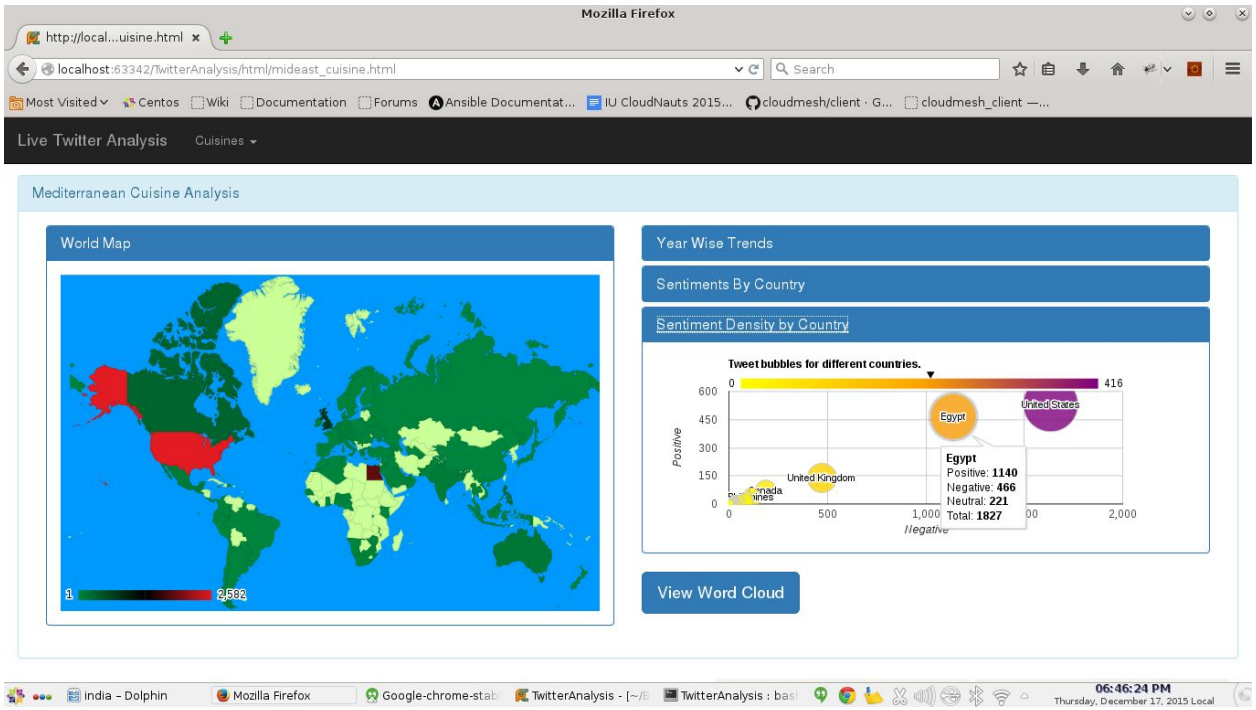


Word Cloud magnifying the most used words:-



Bubble Chart providing a representation over positive, negative axes:-

Total number of tweets from countries determine the size of the bubble.



6. Steps to reproduce

Prerequisite on the host machine to run the results:-

- Python 2.7 (https://www.python.org/downloads/)
- Decent Internet Connection.
- A Github account with the public key added (http://www.wikihow.com/Add-SSH-Public-Keys-on-GitHub)
- Git Client configured (https://help.github.com/articles/set-up-git/)

1. On any machine (local/ remote/ virtual machine) having the above prerequisites, checkout the TwitterAnalysis git repo with the following command:-

```
$ git clone git@github.com:ehdsouza/TwitterAnalysis.git
Cloning into 'TwitterAnalysis'...
Enter passphrase for key '/home/mangirish/.ssh/id_rsa':
remote: Counting objects: 323, done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 323 (delta 9), reused 0 (delta 0), pack-reused 282
Receiving objects: 100% (323/323), 5.03 MiB | 1.61 MiB/s, done.
Resolving deltas: 100% (139/139), done.
```

2. You will see the TwitterAnalysis project directory checked out.
   ```
   $ ll
   total 0
   drwxrwxr-x. 6 mangirish mangirish 54 Dec 17 19:48 TwitterAnalysis
   ```

3. Change your working directory to TwitterAnalysis/html

   ```
   $ cd TwitterAnalysis/html/
   ```

4. We will use the Simple HTTP Server provided by Python to host our web portal:-
   ```
   $ python -m SimpleHTTPServer
   Serving HTTP on 0.0.0.0 port 8000 ...
   ```

5. In browser, hit the following URL:-
   http://<machine_IP>:8000/
   You should be able to view the portal.

Please note that Hadoop HDFS was used as a part of project execution for storing the chunks of tweet data from twitter apis. What we publish in the portal is the processed data which do not require a Hadoop installation.

Note about the Development environments:-
We could host our application locally, as well as in the VM in futuresystem using Cloudmesh. The only problem that we faced when hosting the application on the VM is, we couldn't access port 8000 easily. We added openstack security group rule to allow port 8000, but still the port wasn't accessible probably due to some other intermediate firewalls. Hence as a workaround we could access the VM hosted portal with tunneling port 8000 over SSH.

The VM image that we used is futuresystems/ubuntu-14.04 on the 'juno' cloud in futuresystems.

Git repository wherein the code is maintained:-
https://github.com/ehdsouza/TwitterAnalysis

FUTURE IMPROVEMENTS:

- The analysis can be further extended to various other well known cuisines like Thai, Japanese, Mexican, Burmese etc.
- Make the application dynamic to get twitter feeds and analyse based on keywords of user's choice.