[Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you're ready to add your content, just click here and start typing.]

Trump vs the World

CAB432 Assignment 2
Semester 2 2017

By Joshua Moratalla & Joshua Moratalla

Introduction:

'Trump vs the world' is a cloud based application that outputs a sentiment analysis of tweets directed towards Donald Trump aswell as the user's choice of query. The whole application is divided into three segments; the twitter stream, the database server and the application server. The twitter stream and database servers run in the background while the application server is used by the user.

The usage of the application in regard to the UI is simple; make a query based on tweets containing the user input string & output an overall sentiment analysis of the tweets containing the query string. Not only will it display the overall sentiment, but it will also show a response determined by that aswell as the most recent 5 tweets directed towards trump (@realDonaldTrump). In addition to focusing on Trump related tweets, the user also has the opportunity to look at other tweets.

The background segments, being the twitter stream and database server, handle the retrieval and storage of all posted tweets from the initial execution of the twitter stream. Without these key functions, the cloud application would always have issues having a sufficient number of tweets to analyse aswell as obtaining the tweets in the first place.

In-between the Twitter Stream and the database server, Auto scaling is performed. The Twitter stream auto scaling group requests data from the Twitter API and that data to the RDS instance.

Overall, 'Trump vs the World' is a simple application that uses three servers placed inside AWS cloud infrastructure to function.

Application Description:

Cloud Architecture:

The architecture of the cloud application can be simplified by the image of Appendix [a]. A tweet is firstly sent out from the Twitter API to the first auto scaling instance. From there it analyses the sentiment of the tweet and sends that information to the RDS. Should the average bytes coming out of the instances be over 10, the auto-scaling group will add another instance and continue to do so until the average byte intake goes below the threshold. Then finally, the application server retrieves that data and displays it for the user on a webpage.

Twitter Stream (Auto scaling Group)

The scaling group has an initial instance that performs the twitter stream request and database insertion. It does so by executing TwitterStream.js through the forever module. The JavaScript file makes use of the node-twitter, sentiment and MySQL and forever (continuous stream) NPM modules downloaded through NPM. The JavaScript file itself can be explained simply as a request and insert. It requests all tweets that have at least one English alphabet letter then insert it into the database if the 'lang' parameter is 'en'.

Database Server (RDS)

The database used for the cloud-based application is sourced from AWS as a RDS. It is a simple 20 GB relation database that acts as a storage point for all posted tweets received from the twitter stream. It's the connection between the twitter stream and the Application Server and takes the role as a persistent instance.

Application Server: (EC2 Instance)

The application server is delivered from T2 micro Virtual Machine. It creates the pages by executing a JavaScript file (index.js) through the express module and renders it through the Pug templating engine. This whole execution, however, is performed through the forever module to keep the server accessible at all times.

There are three pages; a homepage, search page and redirect page.

Appendix[b] & Appendix[c] shows the homepage comprising of the average sentiment of tweets with @realDonaldTrump, his response to the average sentiment score, 5 most recent tweets about him and a search section. His responses are based upon the sentiment score from below -2, below 0 and above 0. The most recent tweets shown contain only the time and the tweet text as the layout of the page is very simple and straightforward collectively.

Appendix[d] shows that the search page contains the average sentiment score of the hash tagged query, the 5 most recent tweets about the query, a return to homepage button and obviously another search section, . Both search buttons in the homepage and search page redirect the user to the search page related to their input query.

The redirect page is very simple as seen in Appendix[e]. It tells the user they chose a correct path access the site and redirects them to the homepage.

Overall, the layouts are very simply with complexity only being accounted for in the code.

Scaling and Performance

Scaling on the 'Trump vs the world' application is performed through the Auto scaling group used in AWS. It uses an AMI of the Twitter Stream for replication with an instance range from 1 to 3. The scaling policy used is an average network load of 10 bytes going outwards (instance to RDS). For the load balancer health checks were done by pinging the instance on port 3000 using a TCP protocol.

Performance wise, the selection for the scaling policy worked sufficiently. There could only be 3 instances created at maximum so using 10 bytes would achieve the same 3 instances as if 100 bytes was used. The load would have definitely gone over if the English filter was removed from the stream as can be seen from Appendix[f]. Nonetheless, it shows that instances are created and the application is scalable.

Testing and Limitations

Testing:

Plan	Method	Result	Result Description
MySQL query INSERT	String.replace("'", "")	Fail	Doesn't replace
	String.replace("'", "\'") [1]	Fail	Console.log shows desired output but still has insertion errors
	String.replace(/'/g, "\\'") [2]	Success	Does replace and no issues when inserting into Mysql
MYSQL query SEARCH	LIKE '%query%' [3]	Success	Search for the exact query however it does not perform it with case sensitivity
Persistent Twitter stream effect	The screen function	Fail	Issues occurred during the stream and required a 'manual restart' thus making it inadequate as a continuous stream method
	Forever NPM [4]	Success	Continues the stream and no issues have occurred. Even with disconnection the stream continues.

Limitations:

There were limitations all around however they were mainly due to time constraints as I worked alone during this assignment.

Architecture wise, the comprehension of AWS functions as whole was incredibly difficult. The security groups and channel opening were confusing at times especially when you couldn't perform your own custom RDS security group on its creation. Understanding how the code would perform the auto scaling for the different instances was also difficult.

Load was also an issue as tweets retrieved were in another language thus pointless for the target audience; English speakers. A filter was used where it only retrieved tweets with the

'lang' parameter equivalent to 'en', English. Due to that 2/3 of the incoming tweets entering the twitter stream was ignored. This meant that the load would be significantly reduced hence showing overall reduced load amount than before and less need for scaling. Although that would create less instances, the AWS account only allowed 5 to be used (1 for the Twitter Stream,1 for the Application Server) so the amount of 3 scalable instances was fine.

Another limitation was the coding of the application, a mainly learning Pug templating engine. It was a far better alternative than sending pure html code on responses and could carry variables to create a dynamic page. My poor grasp in Pug was my fault as I had the opportunity to learn it in the previous assessment piece but did not pursue it as there was time constraints in the assessment piece too.

Possible Extensions:

The possible extensions to the application would vary but definitely reside in the realm of twitter.

Tweet About Donald Trump:

Such possible extensions would include a tweet section that allows the user (if they own an existing twitter account) to tweet to Donald Trump so they can see themselves in the most recent tweets. But with that being said, I would have to include user ID in the data the twitter stream would send to the database.
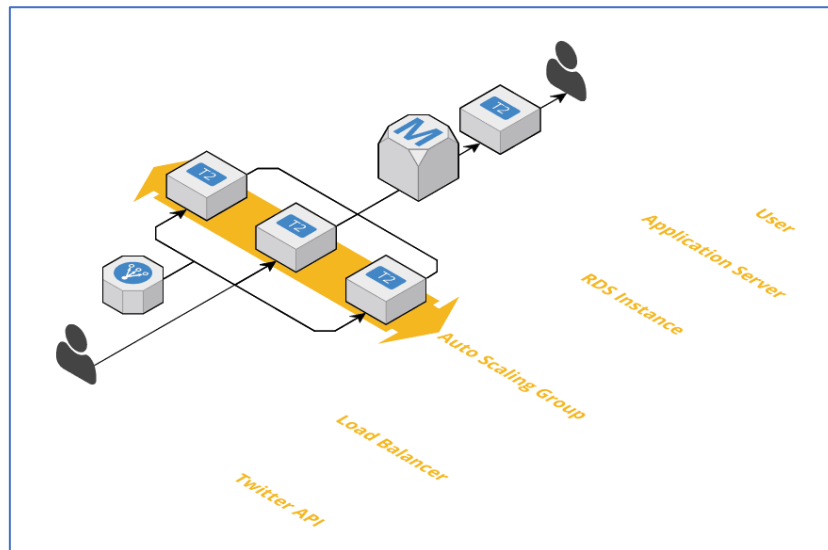
Tweets by Donald Trump:

This extension could allow a user to sign into the application and receive emails about the trending topics that have the words China, Korea, Hillary, Clinton or Trump in their inbox. This would require a sign in function to verify that they do want that email otherwise it would be pointless to send spam to random users from any gathered email.

References:

[1] https://www.w3schools.com/js/js_strings.asp (Accessed 19/10/2017)

[2] https://www.tutorialspoint.com/javascript/javascript_regexp_object.htm (Accessed 21/10/2017)

[3] https://www.w3schools.com/nodejs/nodejs_mysql_insert.aspstackoverflow (Accessed 22/10/2017)

[4] https://www.npmjs.com/package/forever (Accessed 25/10/2017)

Appendix:

Appendix [a]: Cloud Architecture Design

Appendix [b] : Sentiment Analysis tweets directed toward @realDonaldTrump aswell as his response

Appendix[c]: Display of 5 most recent tweets regarding @realDonaldTrump

Most recent Hate/Fantastic tweets about me

Tweet	Date
@realDonaldTrump You never had an original thought in your life, don't try and act like you had one at 6:30, it hap… https://t.co/lGeN5ChwE0	Wed Nov 01 06:26:08 +0000 2017
@BilldeBlasio @realDonaldTrump You've got so many issues u don't control in ur city & you dare to criticize Trump? Ur truly embarrassing.	Wed Nov 01 06:26:08 +0000 2017
@realDonaldTrump But when something like Vegas happens, it's not the time to talk about it!	Wed Nov 01 06:27:52 +0000 2017
@crusher614 @_AmericaFirst_7 @realDonaldTrump Had nothing to do with the false flag of 9/11. Without truth we play… https://t.co/UfOPY4HsPS	Wed Nov 01 06:30:44 +0000 2017
.@realDonaldTrump is no innocent victim of biased media. But that doesn't justify distorted reporting https://t.co/v0Tq5zpHr1	Wed Nov 01 06:31:01 +0000 2017

Appendix[d]: view of search page

Type in to search tweet | Search Tweet

Search Query: #China

Sentiment Analysis: 0.5

Most recent tweets with your query

Tweet	Date
#Nikon to shutter its #Camera plant in #China due to rise of smartphones … - https://t.co/uUbCuoLe3R… https://t.co/pK7T6shfg1	Wed Nov 01 08:14:12 +0000 2017
RT https://t.co/R2RDycGz4C The best view comes after the hardest climb. Do you agree? #China #Travel #Cruise #Yan… https://t.co/gf6yyMr4Sn	Wed Nov 01 08:14:39 +0000 2017
New war thriller https://t.co/LFTbJtvmKO … … available now #ebook #fiction #china #indieauthor https://t.co/7g7ywPDe1w	Wed Nov 01 08:30:08 +0000 2017
Tonight 5:30pm IST #54thABUTV #SongFestival #Chengdu #China. Watch me perform at https://t.co/tWOVui5HiD… https://t.co/ybBcbqdYhg	Wed Nov 01 08:34:17 +0000 2017
China to achieve annual employment target of 11 million new jobs #Chinajobs #TodayinChina #employment #China https://t.co/kPUMsiTK31	Wed Nov 01 08:41:07 +0000 2017

Appendix[e]: Redirect Page saying whoopsie doopsie for the user
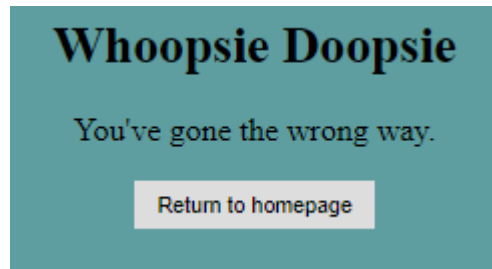
Appendix[f]: flow of traffic from autoscale instance to rds