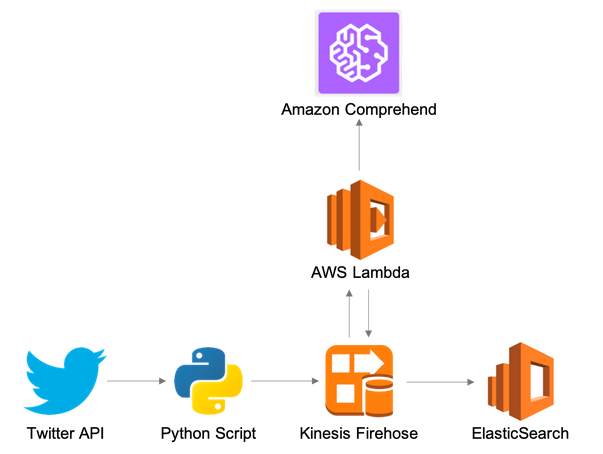
**Data Pipeline: Twitter Sentiment Analysis**

# Work Flow:

### Twitter -> AWS Kinesis Firehose -> AWS Lambda -> AWS Comprehend -> AWS Kibana

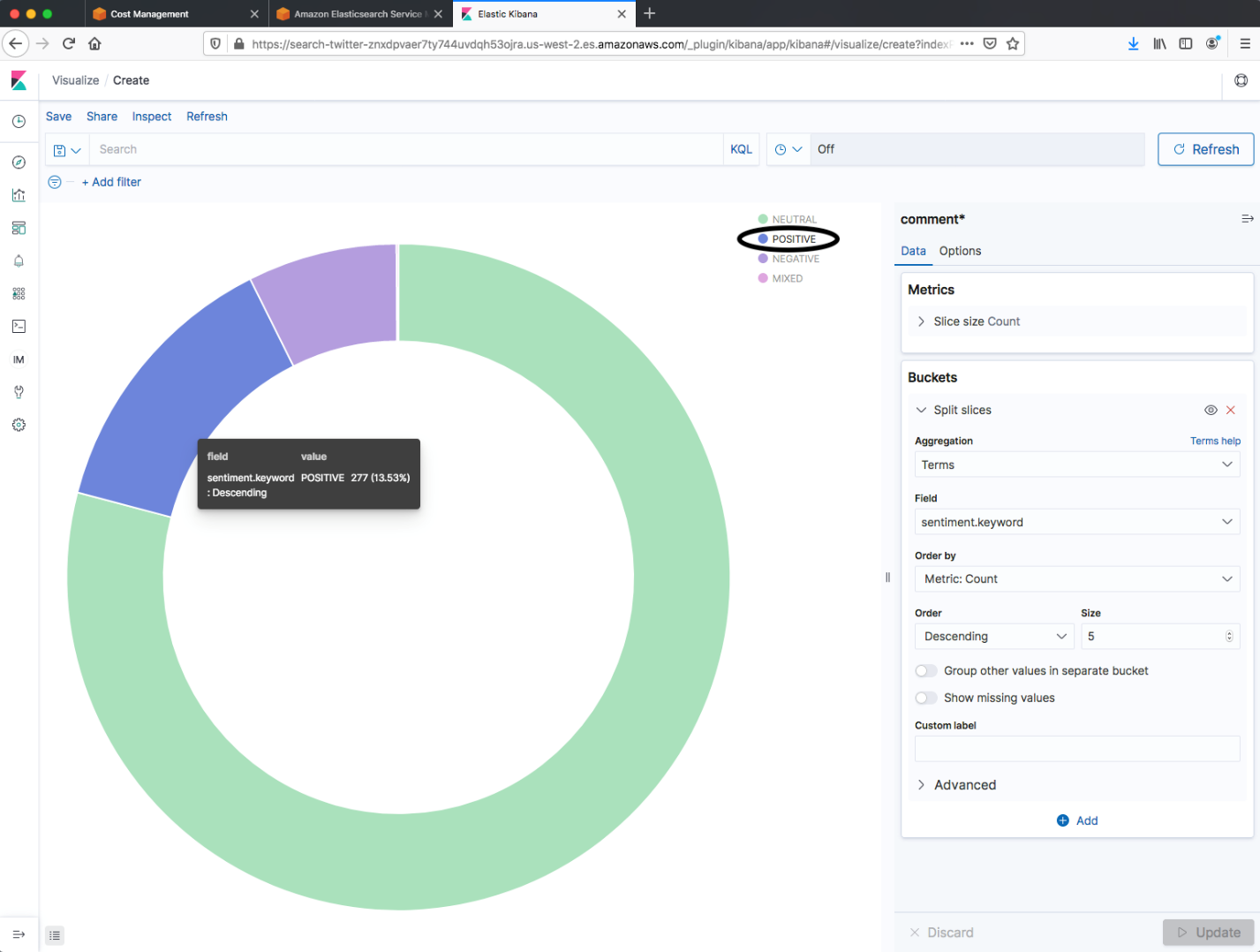
**Architecture:**



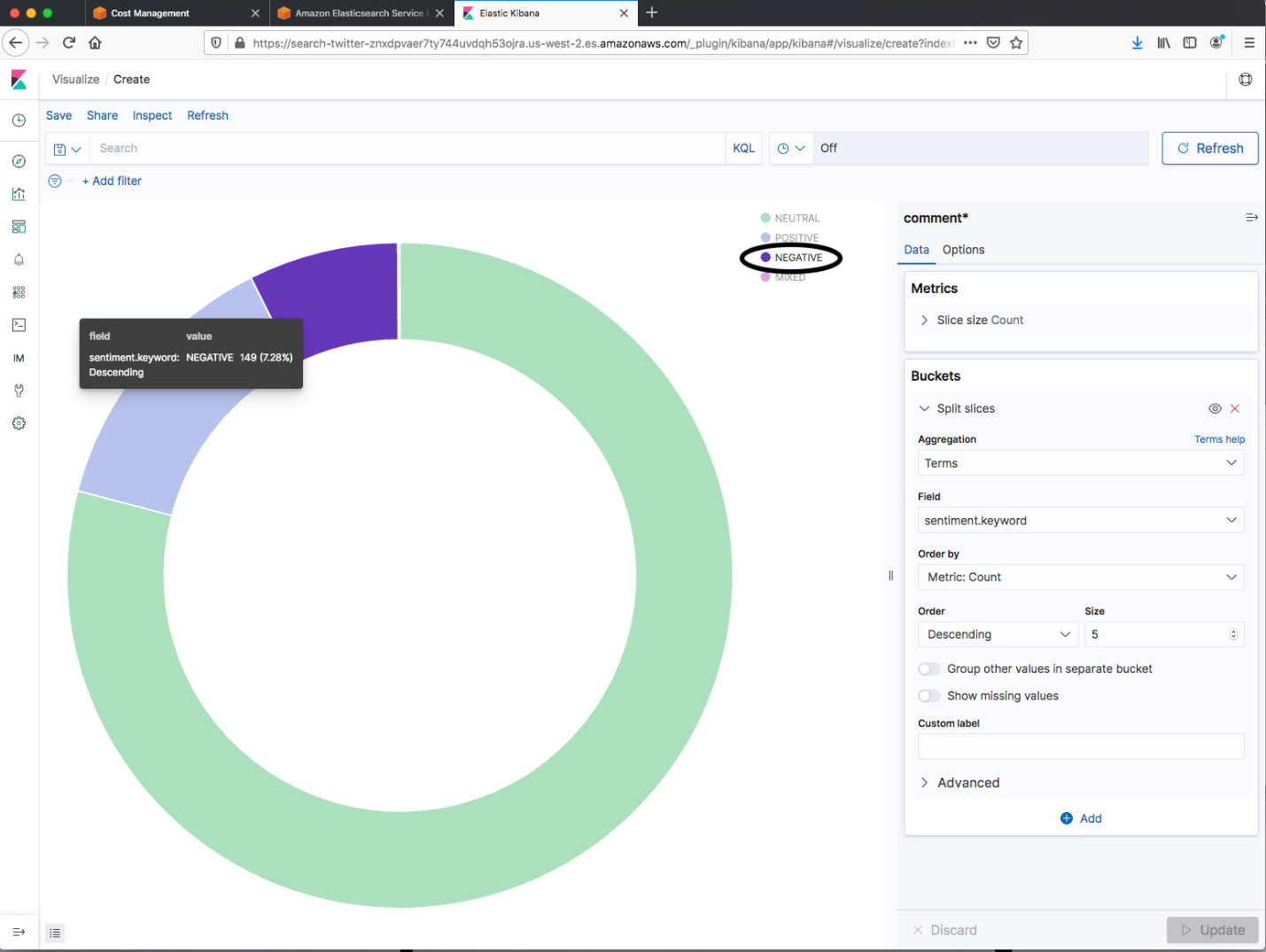
This is a project is on which enables one to stream and analyze the sentiment of live tweets from Twitter based on a certain keyword such as "Bitcoin." It involves setting up a data pipeline between Twitter and AWS Services such as Kinesis, Comprehend and BI visualization tool Kibana. This requires one to have a Twitter Developer account and, of course, an AWS account (Free-Tier). The flow of the project is:

1. Tweets are pulled from the Twitter API after verifying credentials using a Python code running on AWS Cloud9.
2. The code feeds the stream of tweets into AWS Kinesis Firehose.
3. AWS Kinesis Firehose then passes the tweets to a AWS Lambda function.
4. Lambda, also running a Python code, will then pass the tweets into AWS Comprehend.
5. AWS Comprehend analyzes the tweets and returns to Lambda the sentiment analysis as Positive, Negative, Mixed and Neutral.
6. The sentiment analysis results are then sent back to Kinesis Data Firehose, which will forward the data to AWS Elastic Search and store the tweets in an S3 bucket.
7. The data is visualzed on AWS Kibana as the following:

Positive Sentiment:



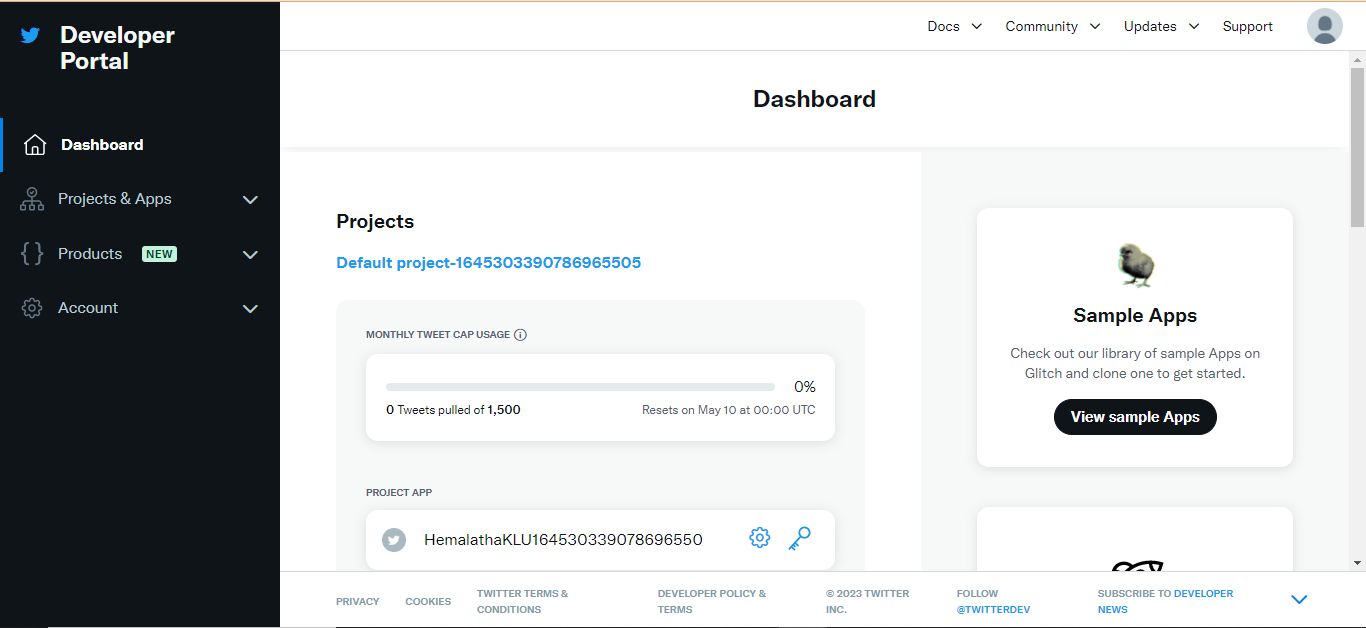
Negative Sentiment:



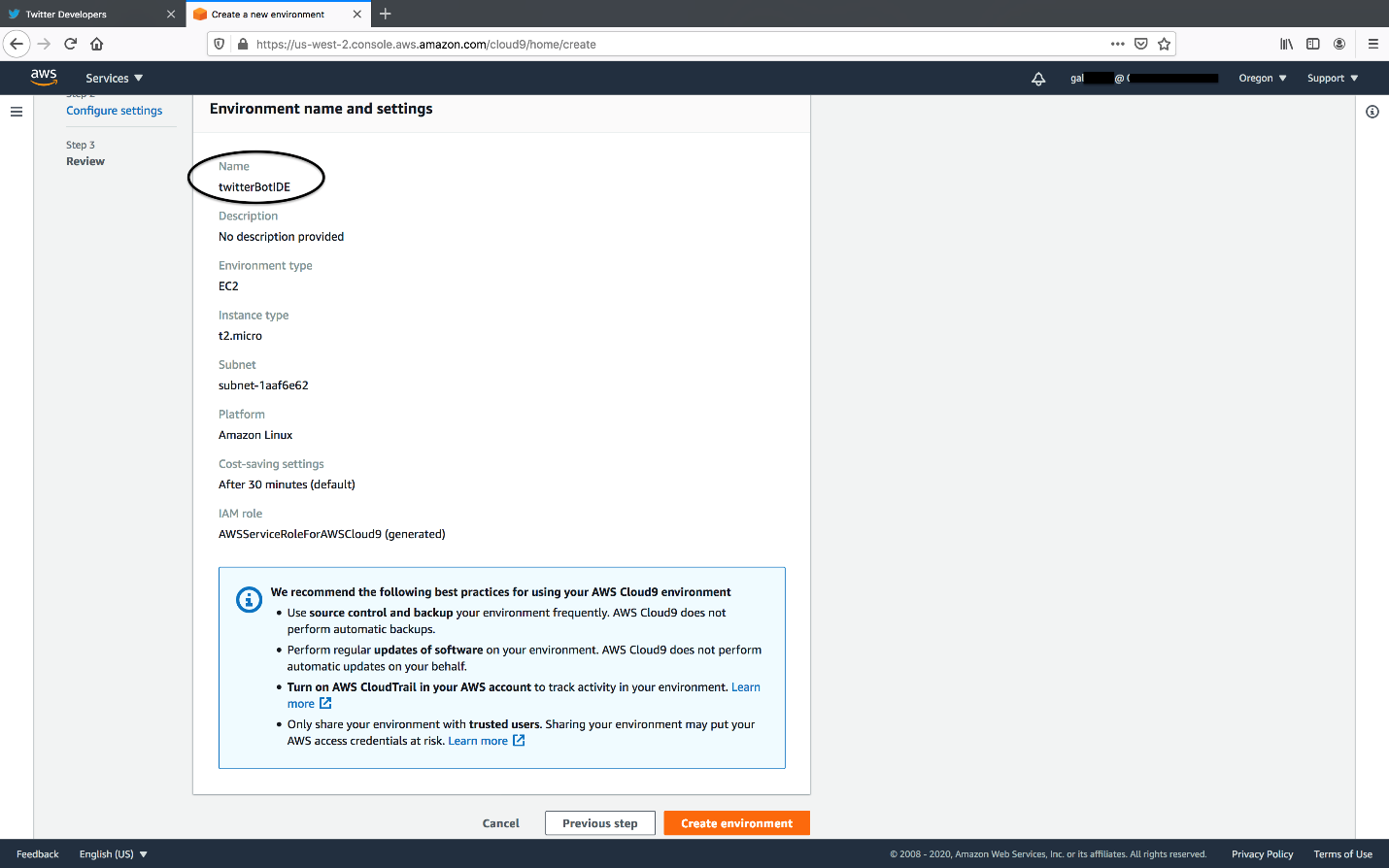
### STEPS:

* Please note that the following steps assume that you have a basic working knowledge of AWS Console and services such as Cloud9, Lambda and Elastic Search. Also, please note that although this project is run on AWS Free Tier Account, leaving services running (such as Elastic Search) will incurr costs. Thus, it's important to turn off services after testing to avoid charges.

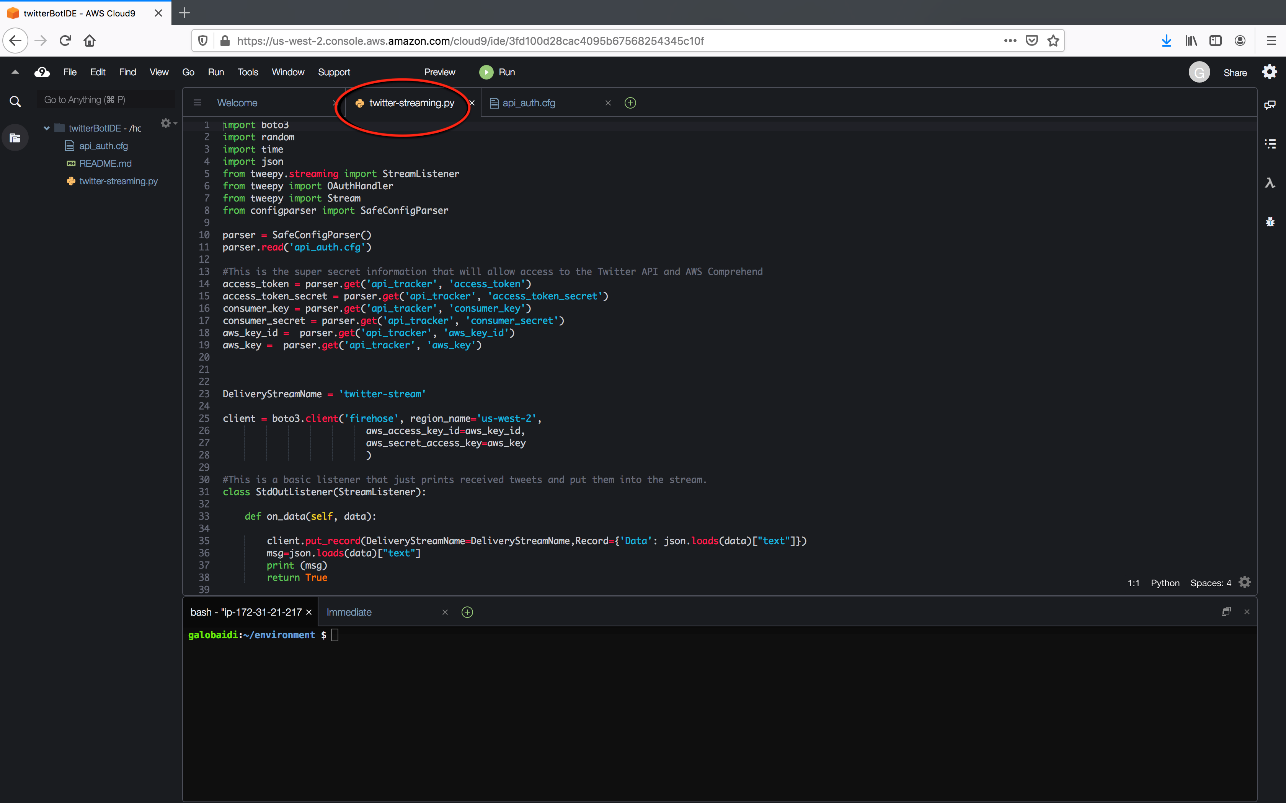
1. Sign up for a Twitter Developer account via <https://developer.twitter.com/en>
   * This step is important in order to get access to the Twitter API (500,000 Tweets allowed per month).
   * Apply for a student version access (answer a few questions regarding your account and usage- in my case it was a student project).
   * Once approved, you will be given a Twitter *API Key* and *API Secret Key*
   * You will then have to a create and name your new App (in this example: *AWSComprehend-GA*) which will be provided with an *Access Token* and *Secret Access Token*.
   * Update the *api\_auth.cfg* file included in this repository with your: *API Key, API Secret Key, Access Token* and *Secret Access Token*



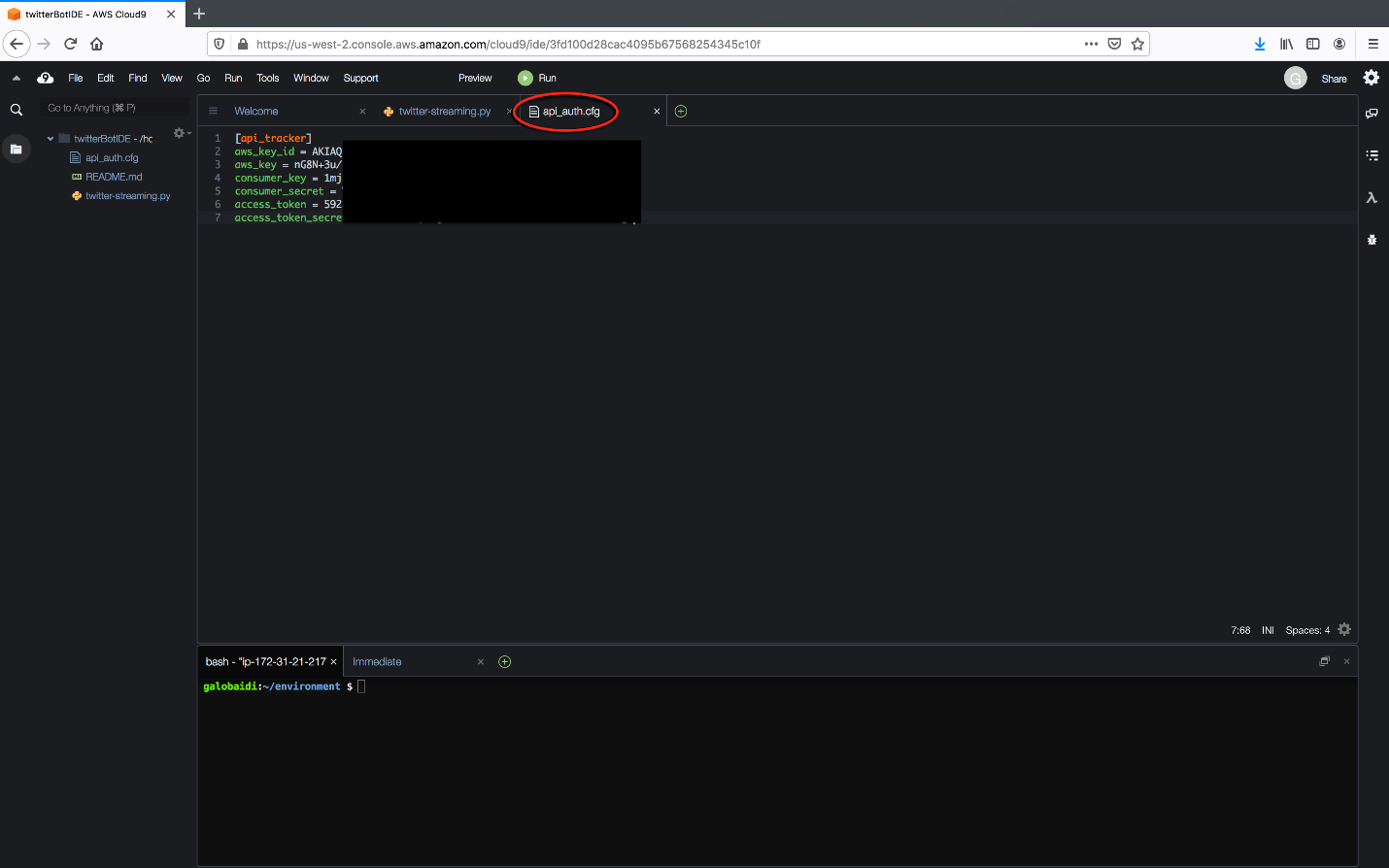
1. Log onto your AWS account using an account that has admin access (preferebly an IAM alias and not your root account).
   * Update the attached *api\_auth.cfg* with your AWS *Access Key ID* and *Secret Access Key*.
   * In order to run this project, certain AWS services could not be executed in Northen California. Thus, I set the region to US-West-2 (Oregon).
2. Create a Cloud9 IDE instance (in this example: *twitterBot*) using default options:
   * t2.micro (1 GiB RAM + 1 vCPU) Free-tier eligible. Ideal for educational users and exploration.
   * Amazon Linux
   * Use your default VPC and subnet.



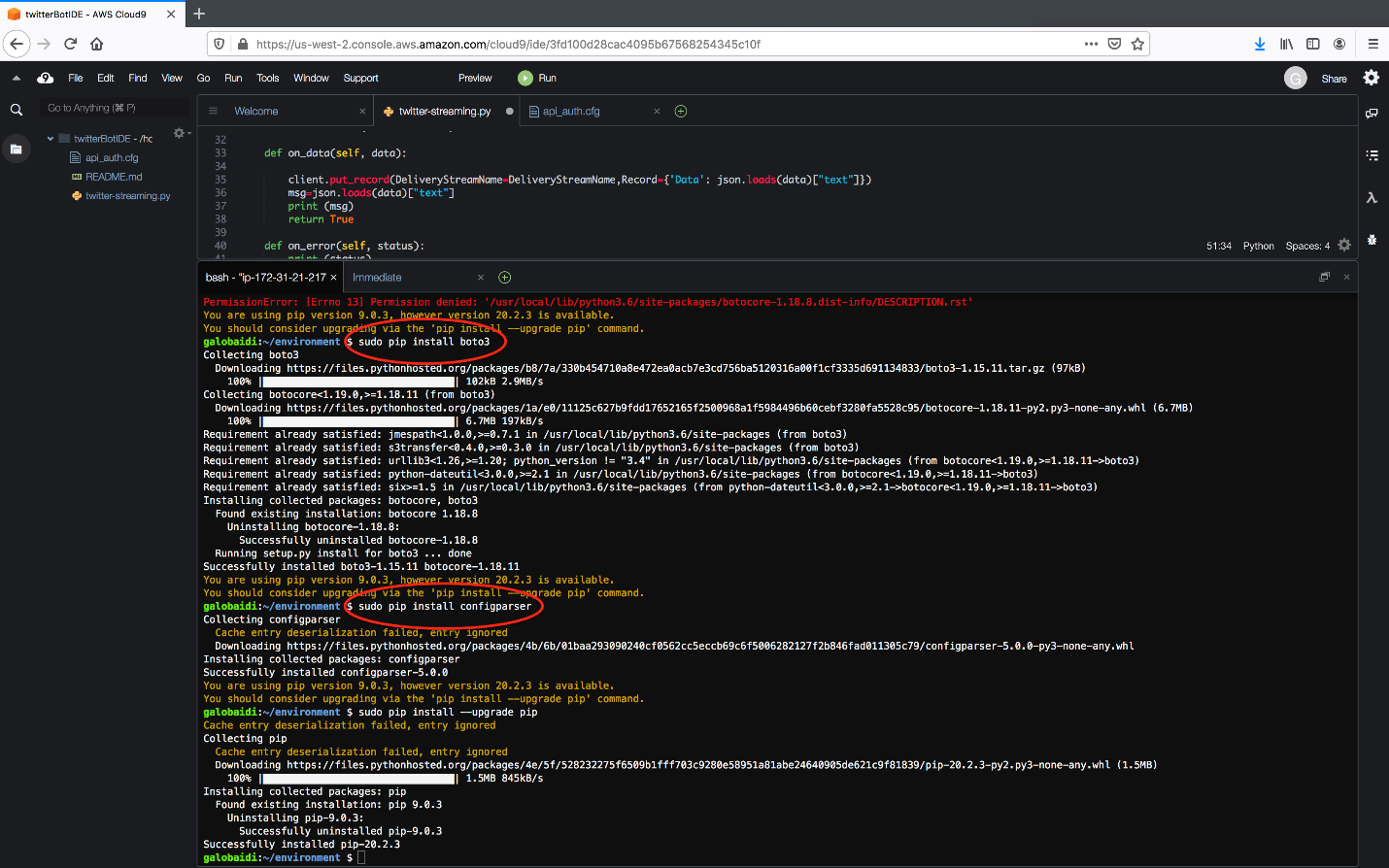
* Create a new Python script in the Cloud9 environment (File -> New File).
  + Download and use *twitter-streaming.py* included in this repository.
  + Note that the variable *DeliveryStreamName* points to 'twitter-stream' which is the Kinesis Firehose that will be created in Step 5.

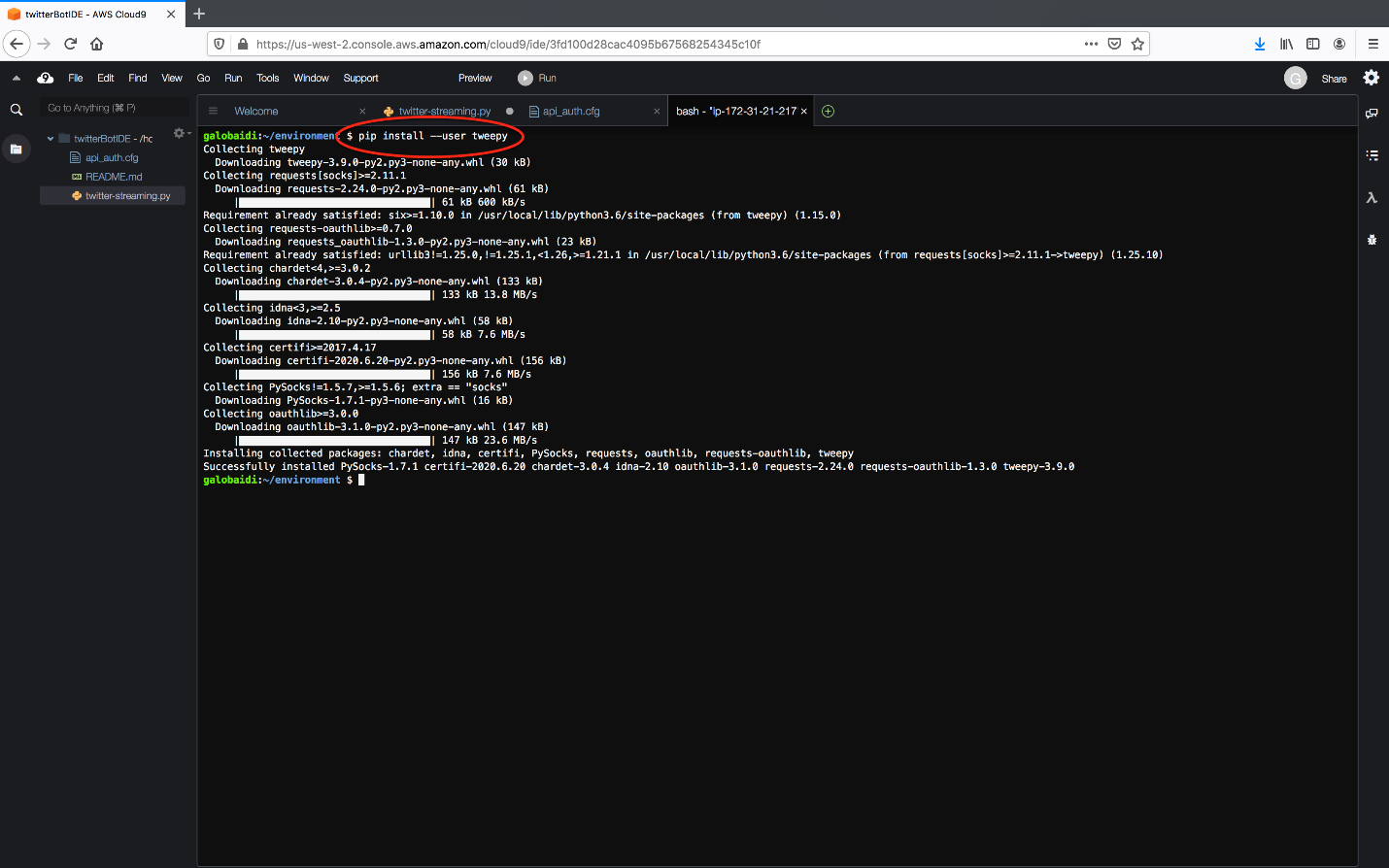


* Create *api\_auth.cfg* file which we have updated in steps 1 and 2.
  + This file holds the credentials that twitter-streaming.py will use to access the Twitter API and AWS services.

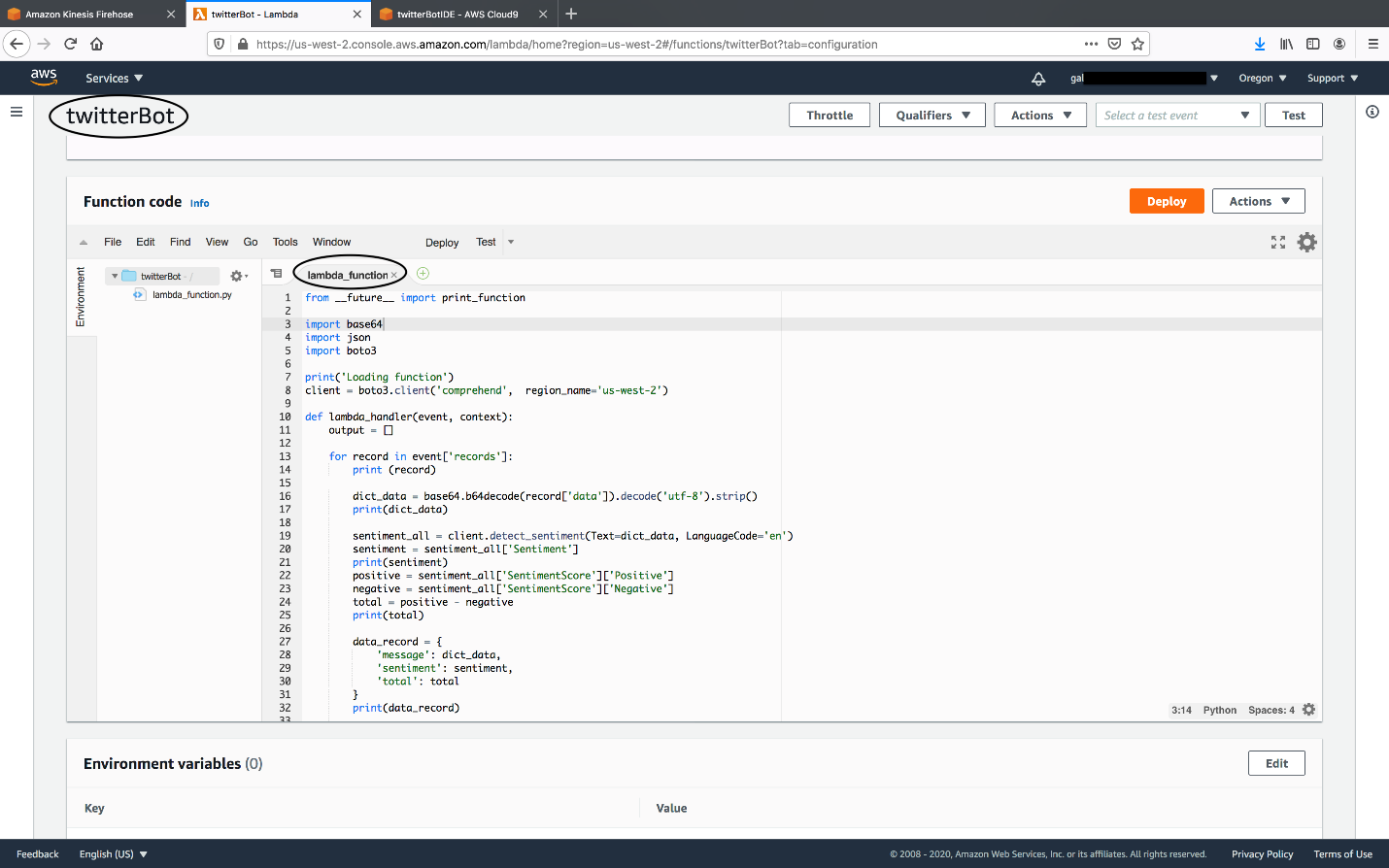


* Use pip to install the modules needed to run twitter-streaming.py:  
  + Install AWS SDK boto3: *sudo pip install boto3*
  + Install configparser: *sudo pip install configparser*
  + Install Tweepy  
    - Close and open a new bash terminal in Cloud9 in order for this to pip to work
    - *pip install --user tweepy*

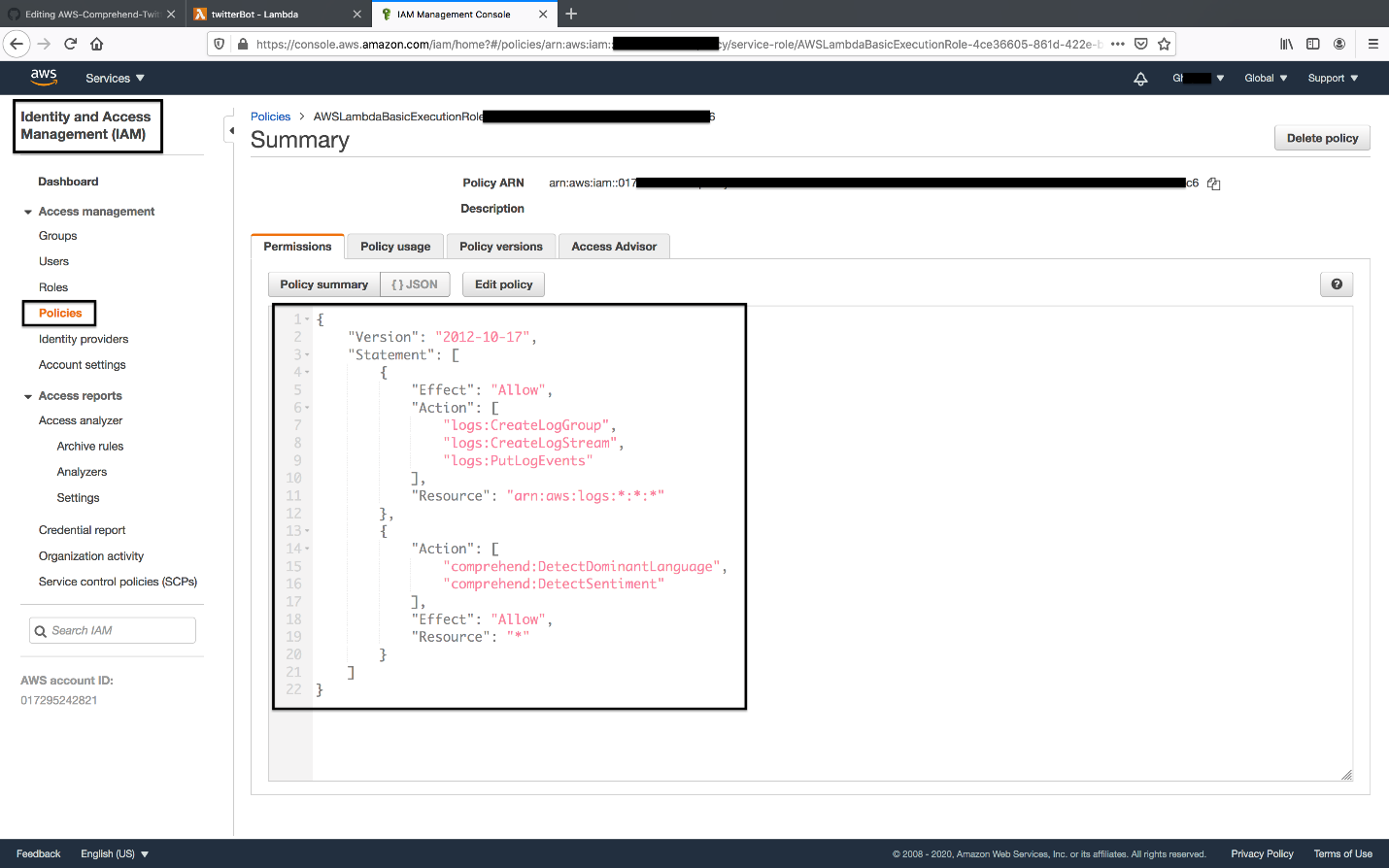


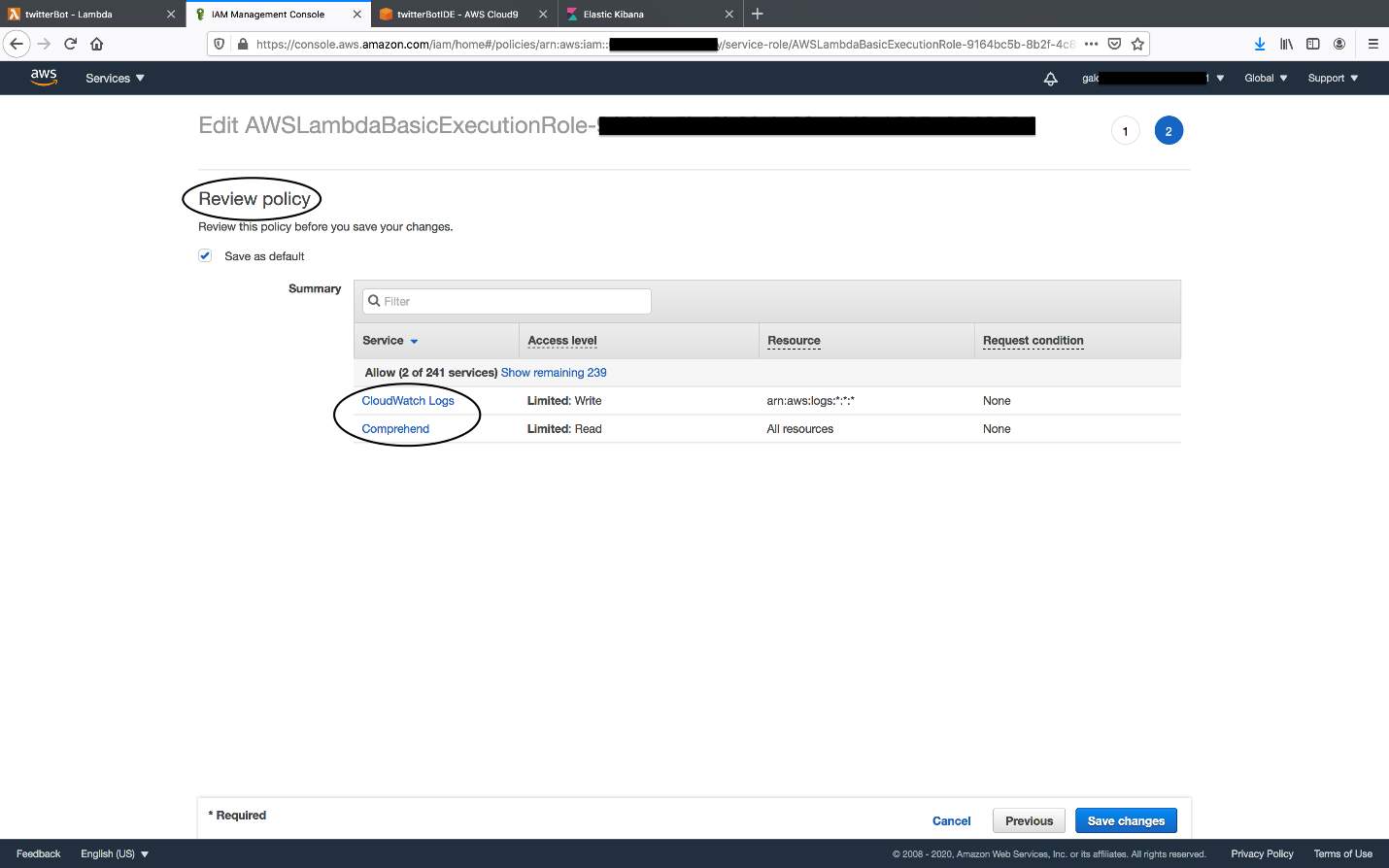


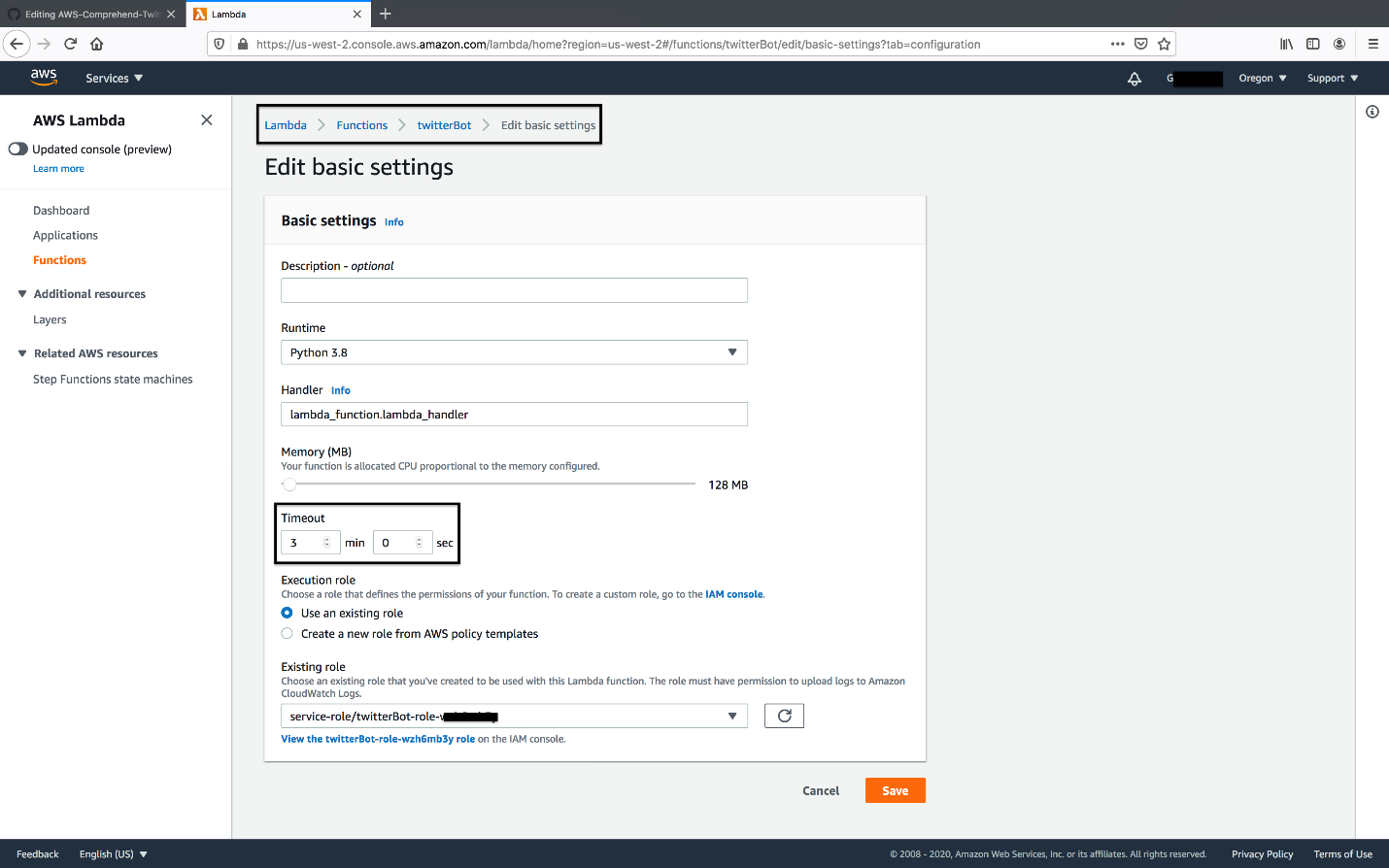
1. Create a new Lambda Function which will receive the stream of tweets via Kinesis Firehose and then send to Comprehend. Use the following configuration:
   * Choose the option: Author from Scratch
   * Name the function the same as the Cloud9 instance name set up in Step 3 (in this example: *twitterbot*).
   * Runtime: Python 3.8
   * Download *lambda\_function.py* attached in this repository and enter as the code for your new Lambda function.



* After the function is created, Go to Permissions -> edit Role name
  + Edit the Lambda's policy JSON so it will have access to Comprehend with the following configuration:

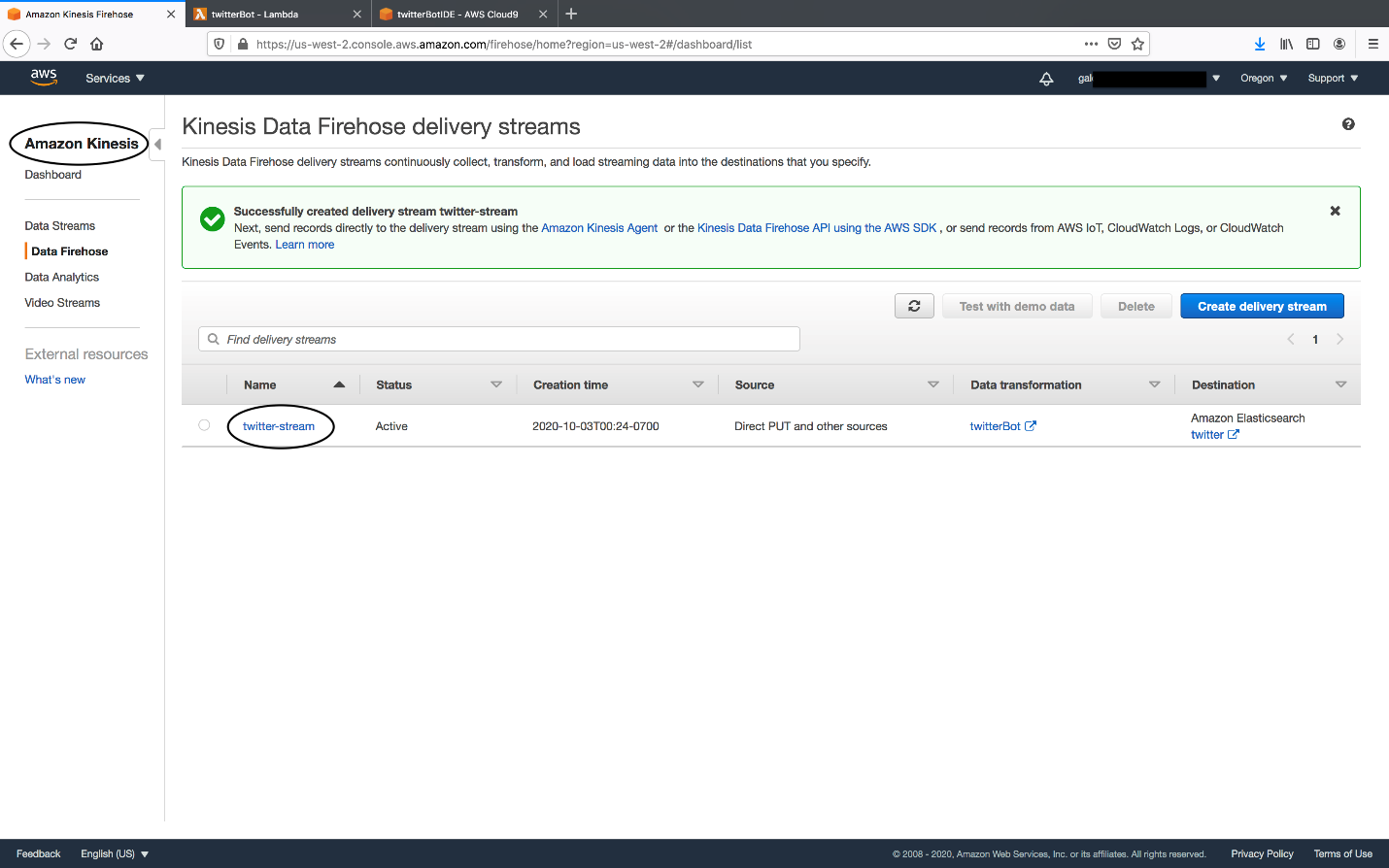


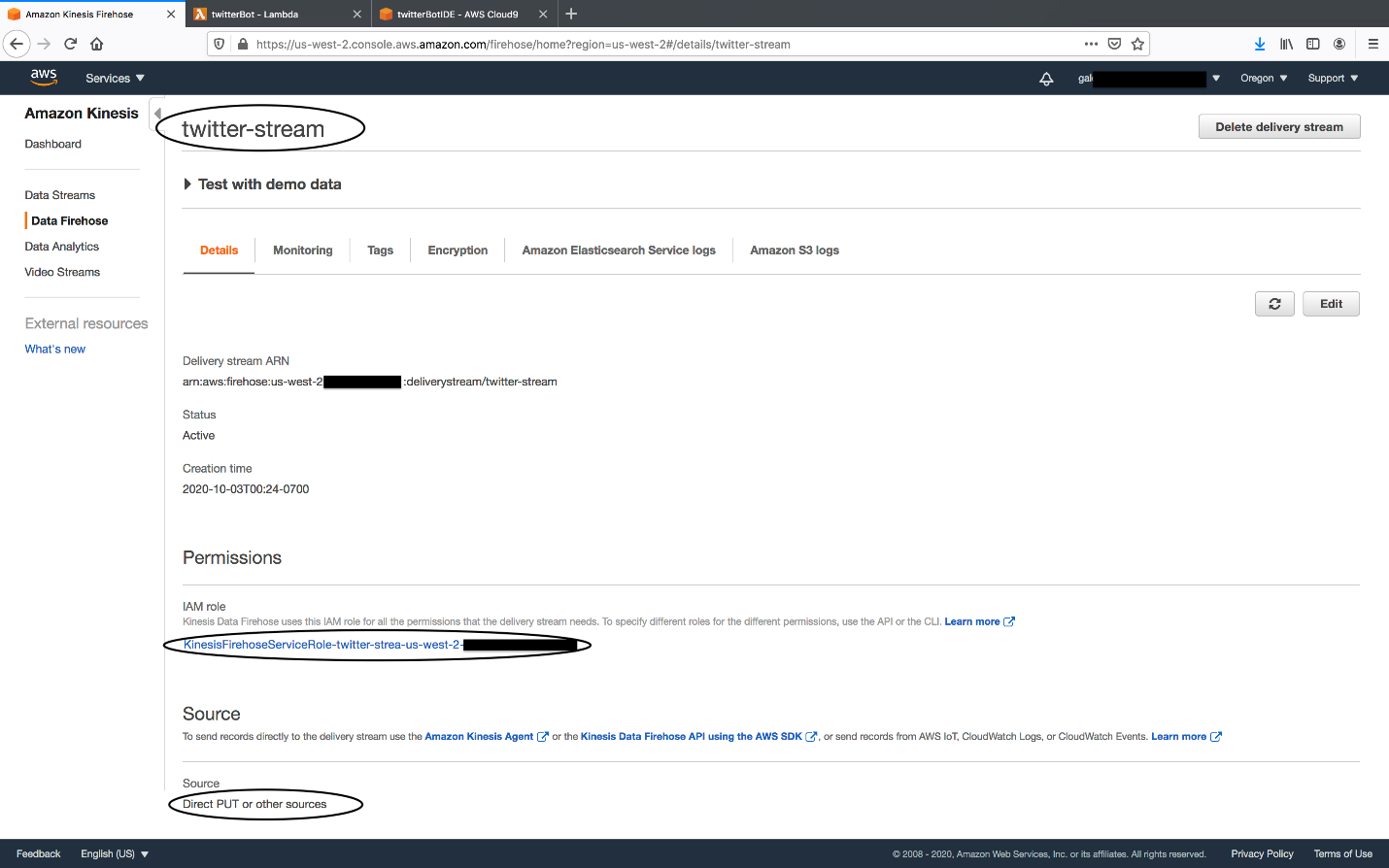




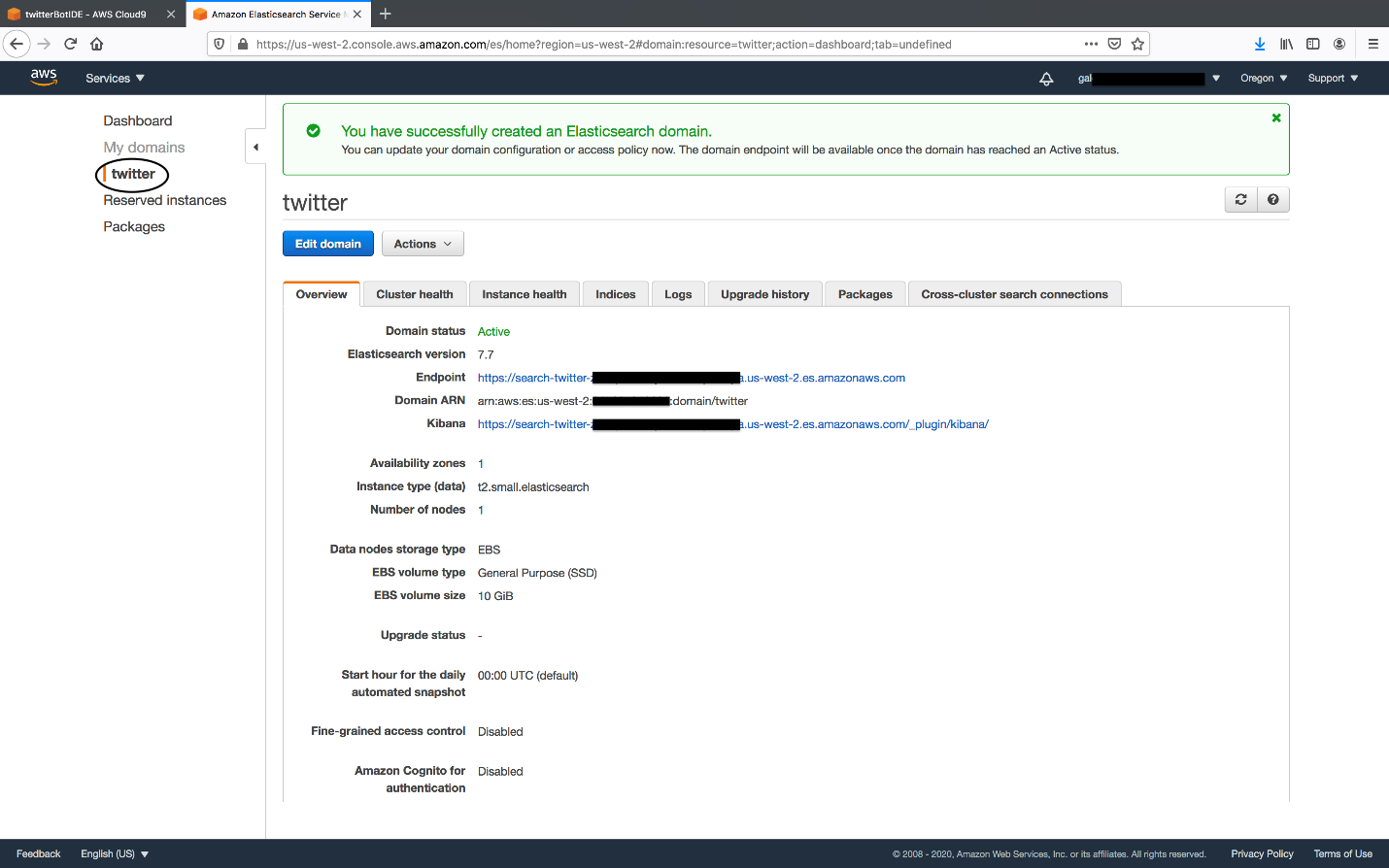
Setup AWS Kinesis Firehose:

* If you recall in Step 2 in streaming-twitter.py, there is a variable in the script called *DeliveryStreamName*. The variable points to *twitter-stream* which is the same name that we are going to name this Kinesis Firehose. Otherwise, the Python script in Step 2 will not work since it will call on a Kinesis stream that is nonexistent or with a different name.
* Use Lambda function in step 4 as its "Data Transformation.

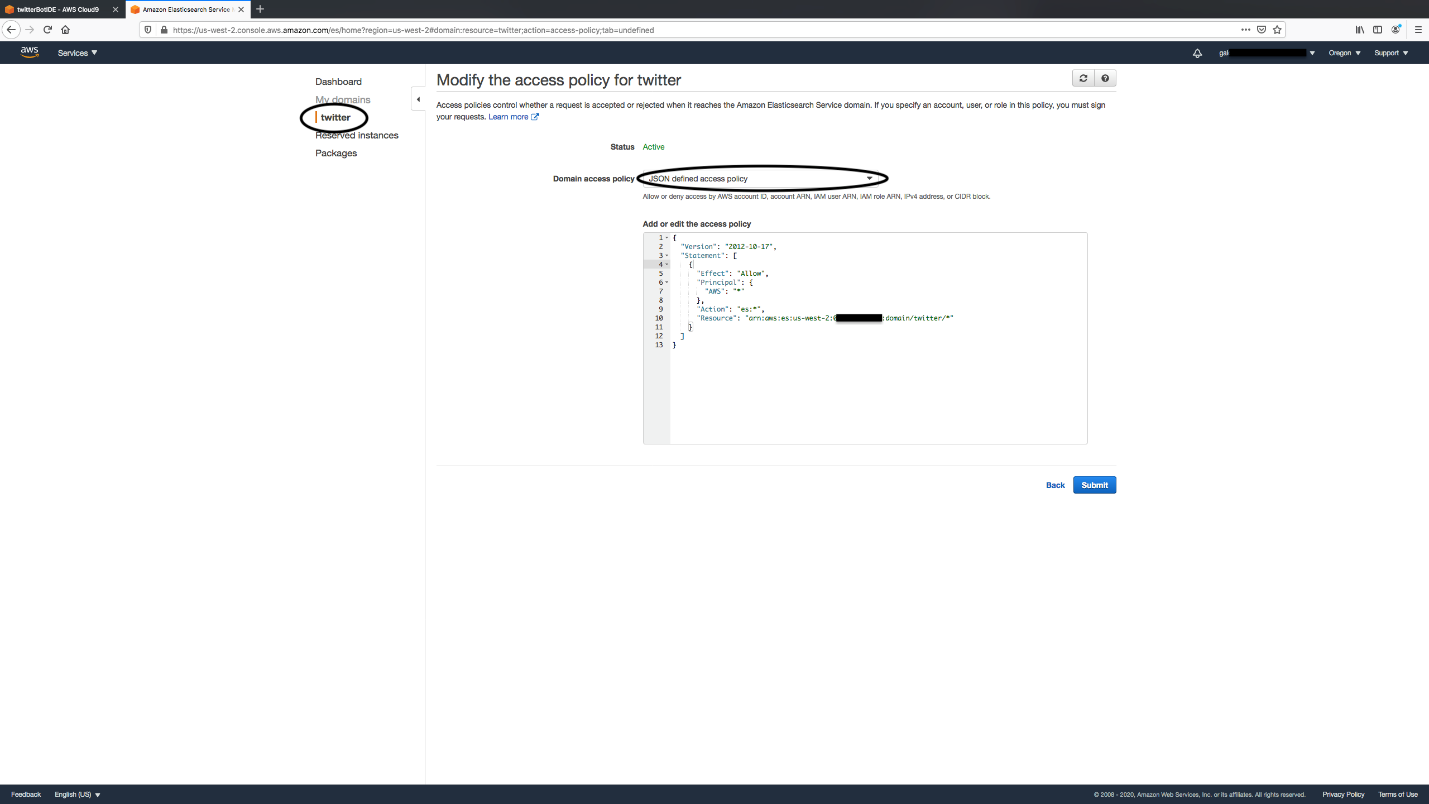




1. Setup AWS Elastic Search domain & Kibana (in this example: "twitter") which will receive the sentiment analysis from our Kinesis Firehose. Use the following configuration:
   * t2small instance
   * For the sake of simplicity in this project, we will setup a public domain access policy: *iPv4 address : \**and we will uncheck *"Require HTTPs for all traffic to the domain."*  
     - Note: This is not recommended for future setups but once again we're doing this for a quick test in this project.



* Once active, we will modify Elastic Search domain *twitter's* access policy so it can access Kibana. Kibana will be used to visualze the results.
  + Please remember: leaving this service running will eat up your Free-Tier usage and you will incurr costs. Stop instances and services after running this test.



1. Run twitter-streaming.py script on Cloud9:
   * Execute the code in Cloud9 bash terminal by typing the command *python streaming-twitter.py*
   * Run the script for about 15-30 minutes to collecte data on the provided keyword.
2. Launch AWS Kibana and analyze the collected data:
   * Please see images of the Pie Charts created for Postivie, Negative, Neutral and Mixed sentiment anaylsis at the beginning of this file.