**Detail Report**

**Topic:** Demonstration of Azure Face API to collect face attributes from face images

**Problem:** To collect data intelligence from people’s faces , their expressions and kind of attendance for a specific public gathering/event with use of Azure Cognitive Services - Face API, Azure Storage and Visualization with Pandas.

**Data Set:** Input Data source is, JPG Image database from Georgia Tech Face DB - <http://www.anefian.com/research/face_reco.htm>

Database has multiple color images (127 MB in size) of different people in variety of lighting conditions. The data is publicly available for research purpose. I used a manageable subset for purpose of this demo and used them as input to my program as a directory location.

**Hardware:** Intel i5-6300U CPU 2.4Ghz, 16 GB RAM, 64 bit Windows 7 OS

**Software:**

|  |  |
| --- | --- |
| **Technology / Tools** | **Description** |
| Azure Cognitive Services – Face API | Azure’s Face API services |
| Azure Storage - Table | Azure’s Storage services - Table |
| Python 2.7 & 3.6 | Python 2.7 & 3.6 (Used 2.7 for Visualization due to 3.6 installation issues) |
| Microsoft Azure Storage Explorer | To manage Azure Table data |

**Result Analysis:**

In test demo I ran, program picked up all images from input directory to process them as Batch. Azure Face API was called per image to detect face attributes. Though Face API collect a bunch of attributes, I have limited results to Age, Gender, Glasses and Smile attribute for purpose of Demo. Once results are retrieved, attribute values are stored in Azure Storage Table. I created a Data Visualization code using Pandas and gave extract from stored face attributes from Azure table to find average age of all faces and average smiling face attribute of this test group to display as graph. If we consider this input source as set of people attending a public event, we can see average age of attending group and whether they were smiling or not in event. Including other attributes, such results can provide public reception of any social events.

**Lessons Learnt :**

* There are multiple image data sources available to test with but lot of images are in non JPG format mostly RAW format. I could not get Python packages working with non JPG format images. I also saw that Azure Face API only works with JPG, BMP, PNG and GIF files.
* Azure Face API (Free pricing Tier) can only run 20 calls per minute. So when I was running for big batch, I was getting errors. I reduced down my input to < 20 for Demo. We can use time delay function to overcome this limit to some extent.

**Pros :**

1. Azure Face API accurately delivers face attributes and specially age
2. Face API can detect very small faces in images correctly. In one of my test image, it captured a face in picture on shirt person wearing in image
3. Free pricing Tier is available for use

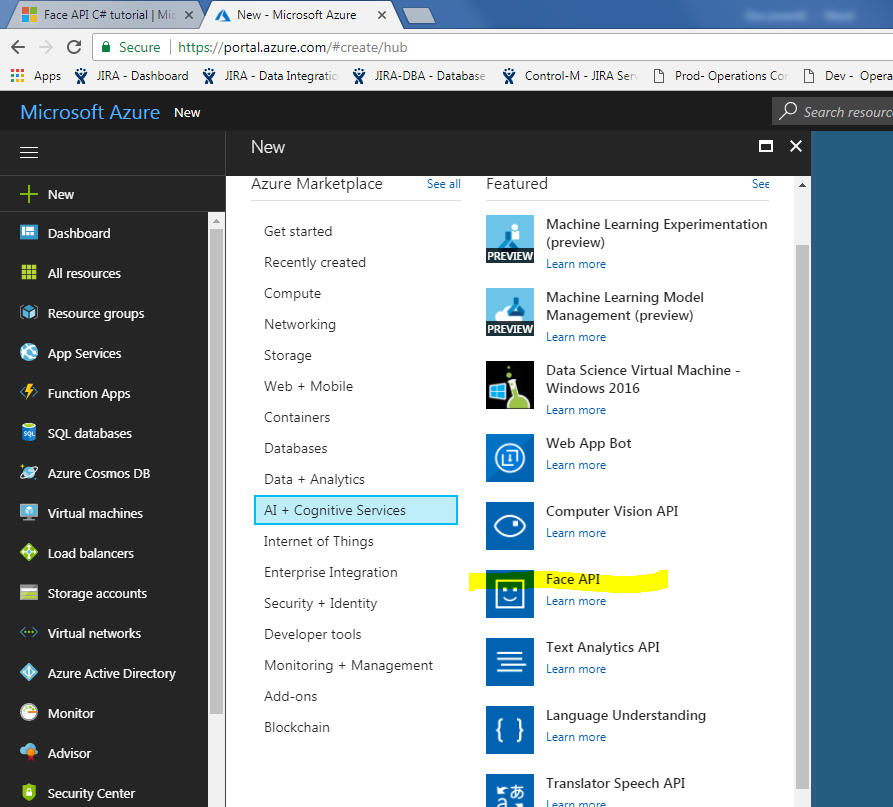
**Cons:**

1. Azure Face API does not work with RAW image formats produced by cameras. It only works with JPG, BMP, GIF and PNG
2. Free tier has limitation to run 20 calls in a minute

**Steps followed in Detail:**

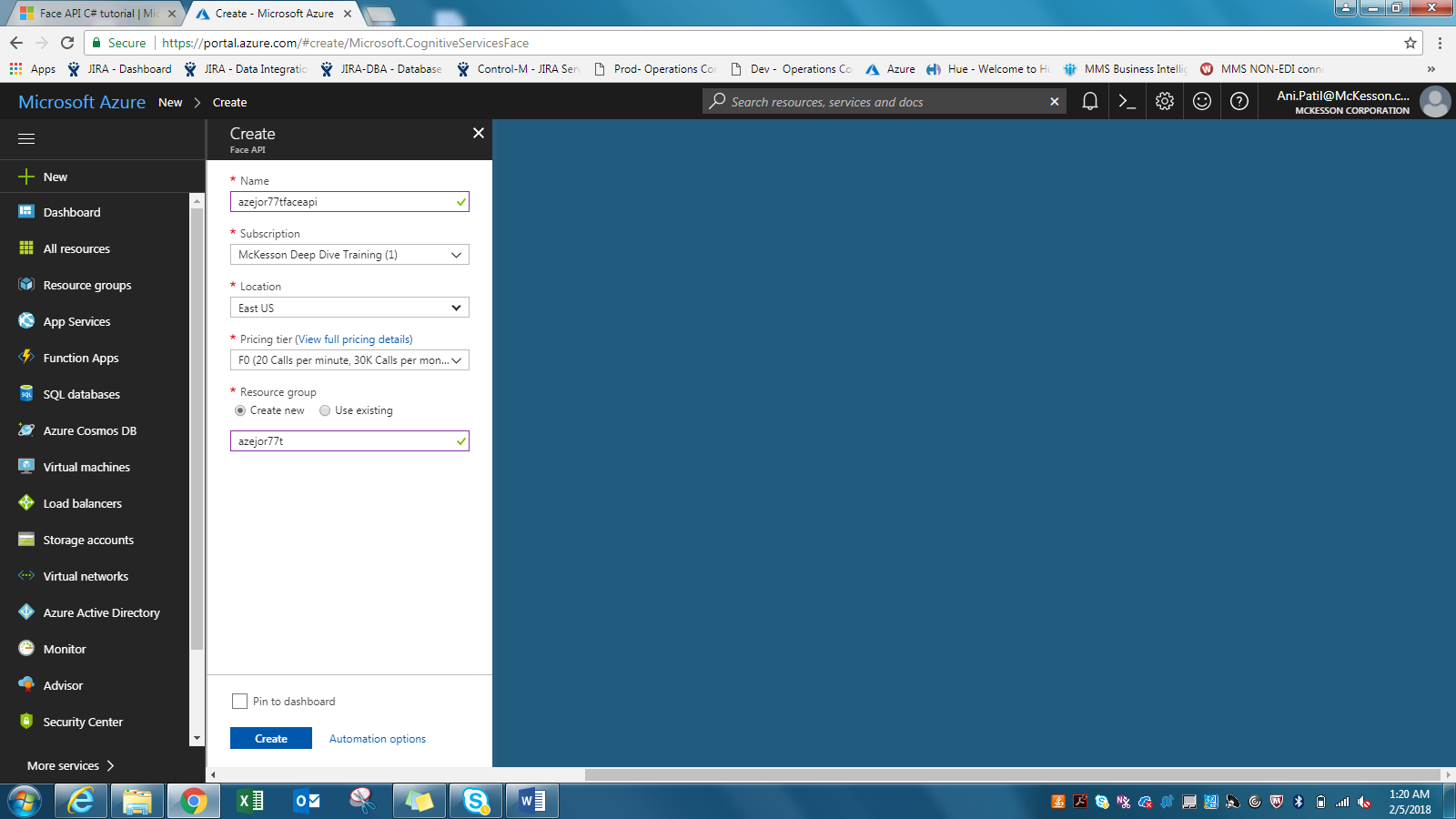
All steps for installation and configuration of software are given below-

First login to Azure Portal and Subscribe to Face API

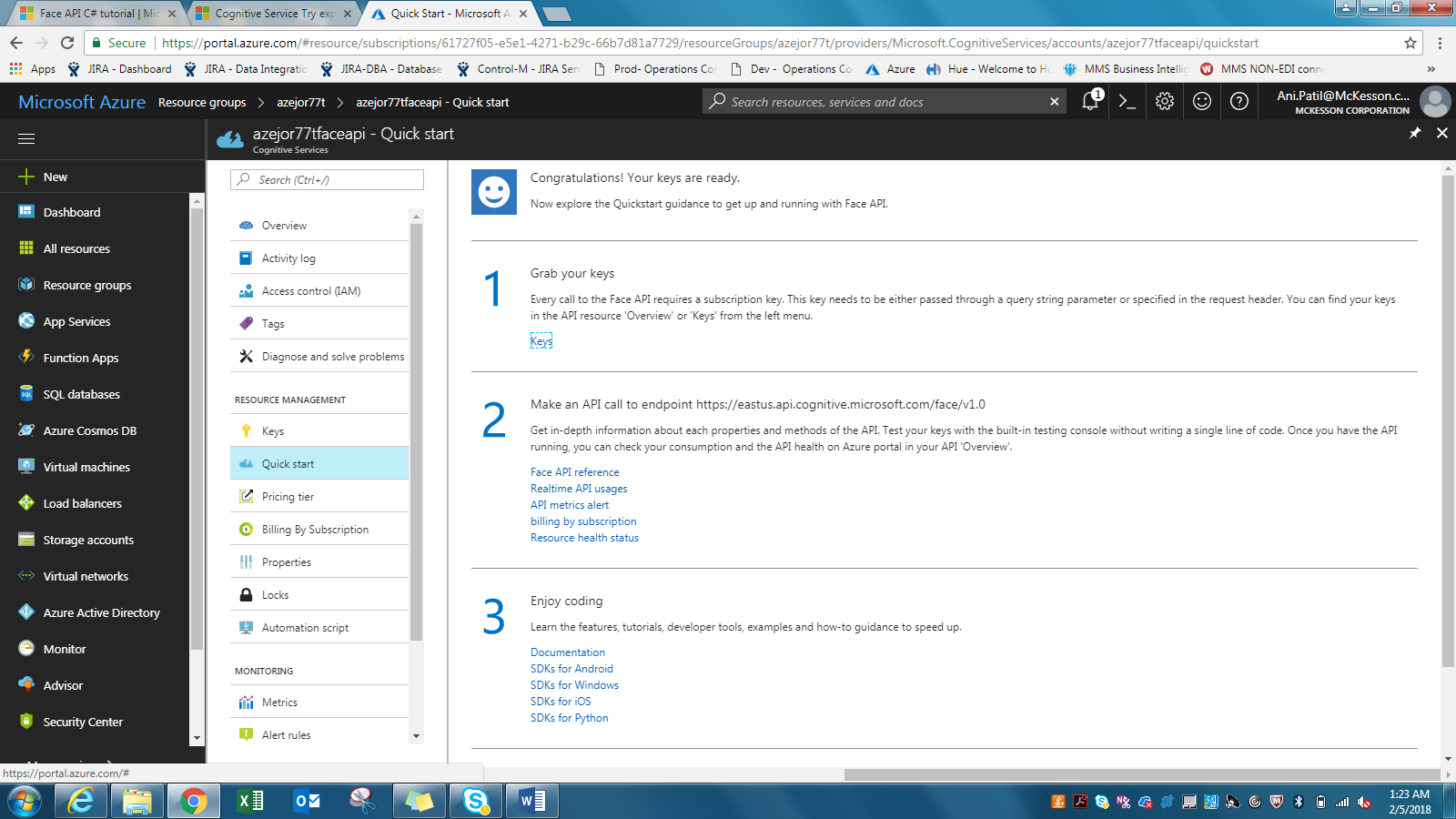


Create Face API.

Below see that we have selected Free pricing Ties with 20 calls per minute limit.



Get your Keys



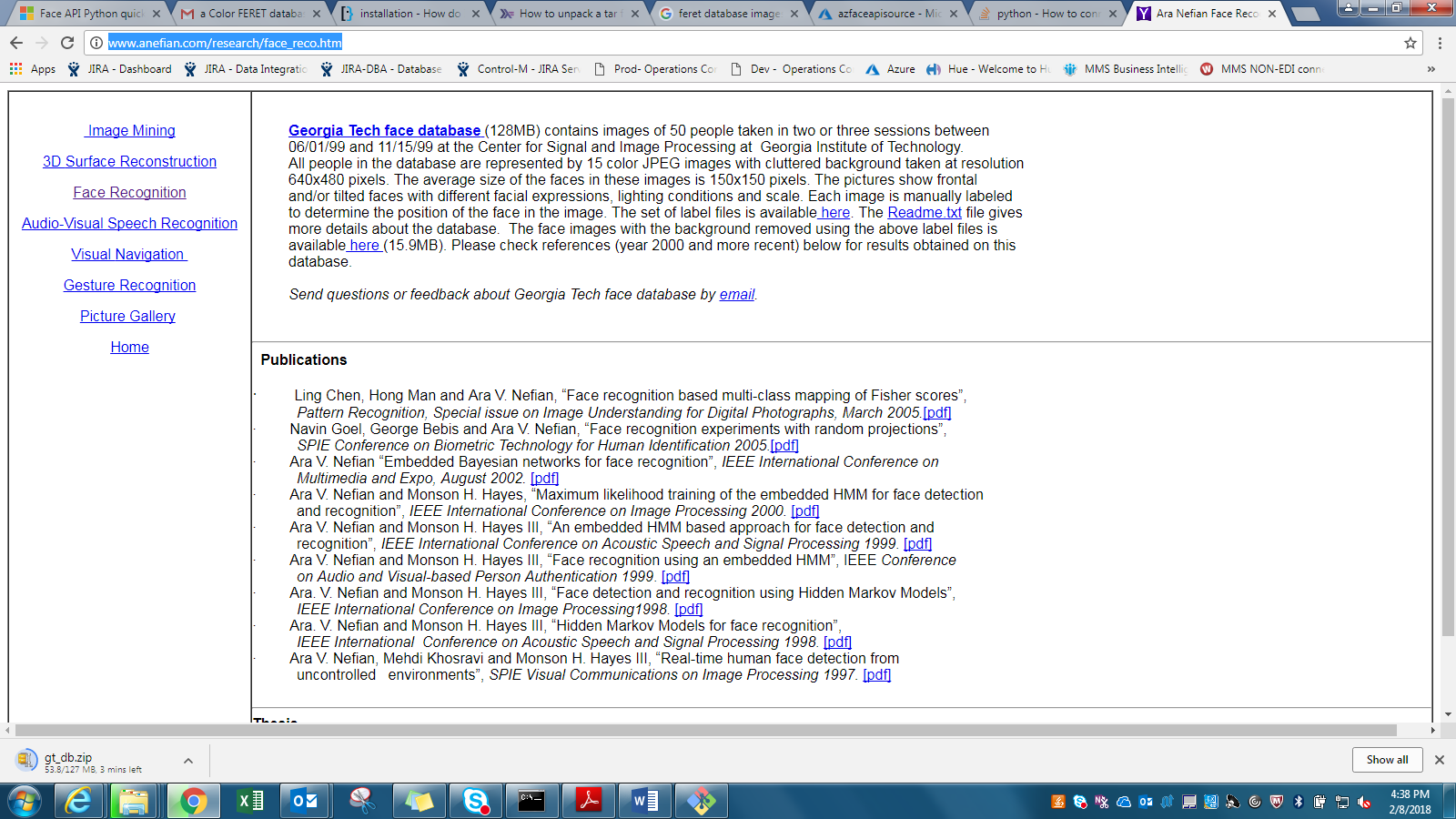
Key is - xxxxxxxxxxxxxxxxxx

Now lets download input image source database.

Image database

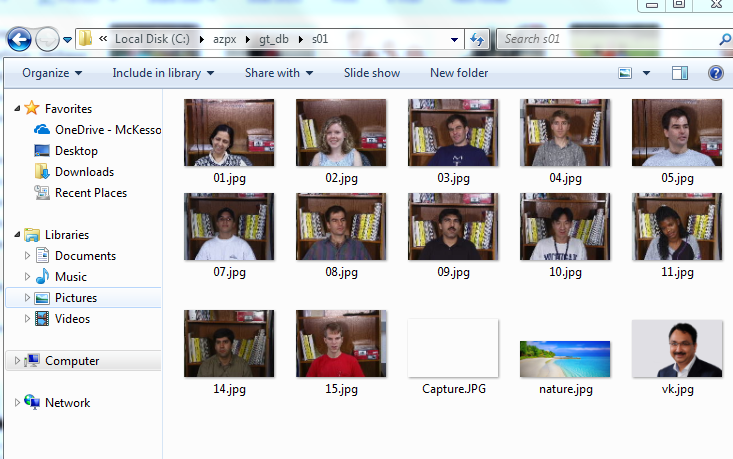
<http://www.anefian.com/research/face_reco.htm>

To download, click ‘Georgia Tech face database’ below



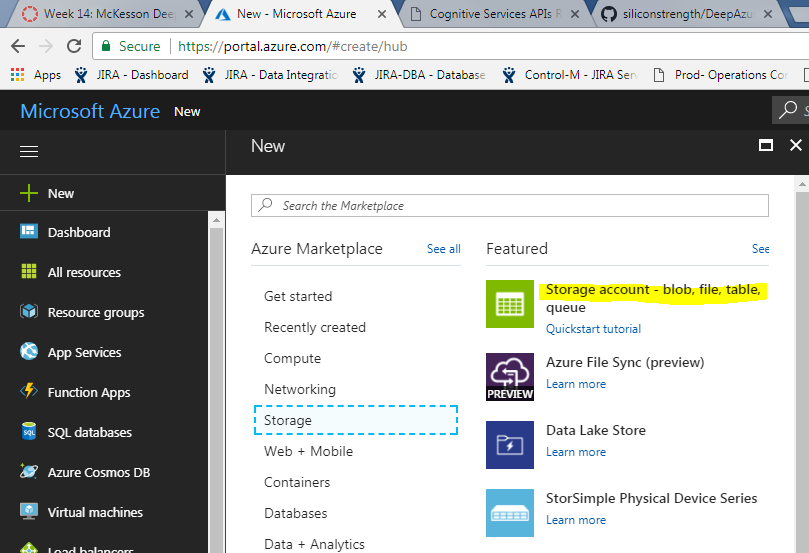
This Database has JPG images of multiple people in different lighting conditions. Complete database is about 128MB in size.

Download image database and take subset of images in a directory for Project demo. Below is a subset I used for this demo.

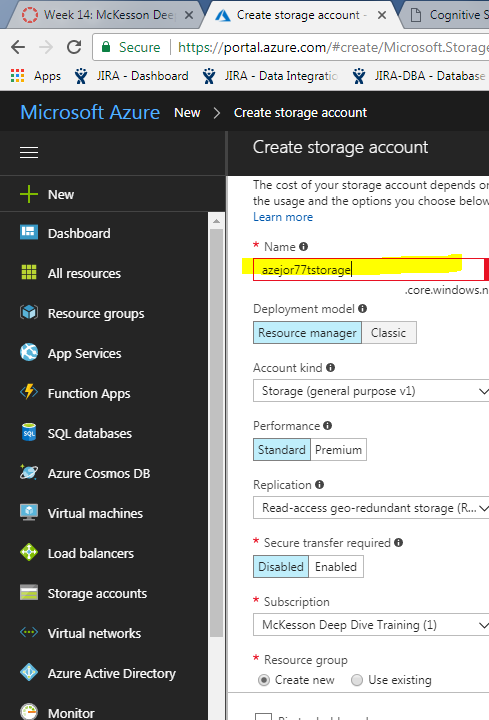


If you see above image of input images, I have added some random pictures which does not have any face in them. This is to test if Face API identify it correctly to not detect faces.

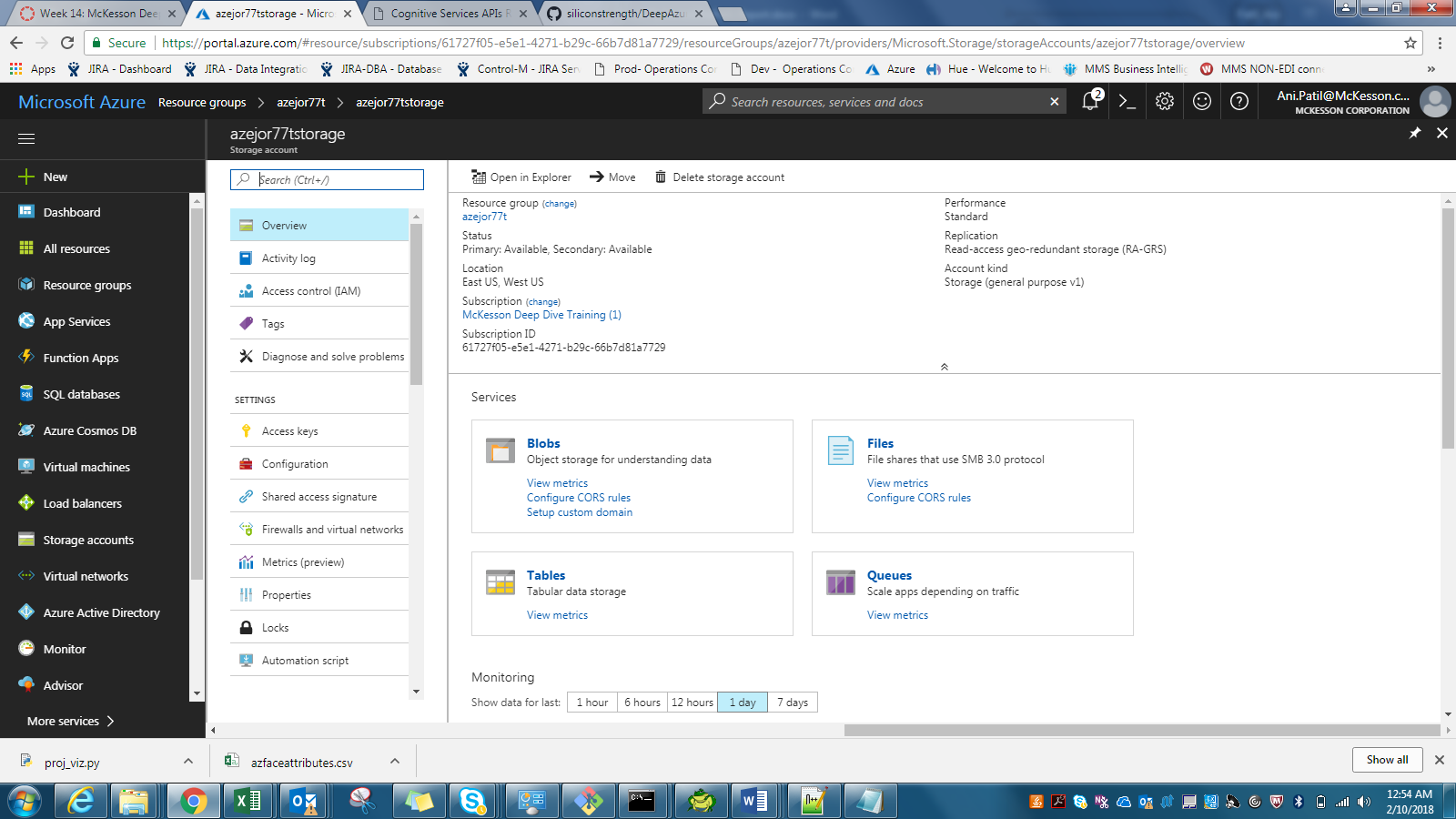
Now we will create a new storage account on Azure portal



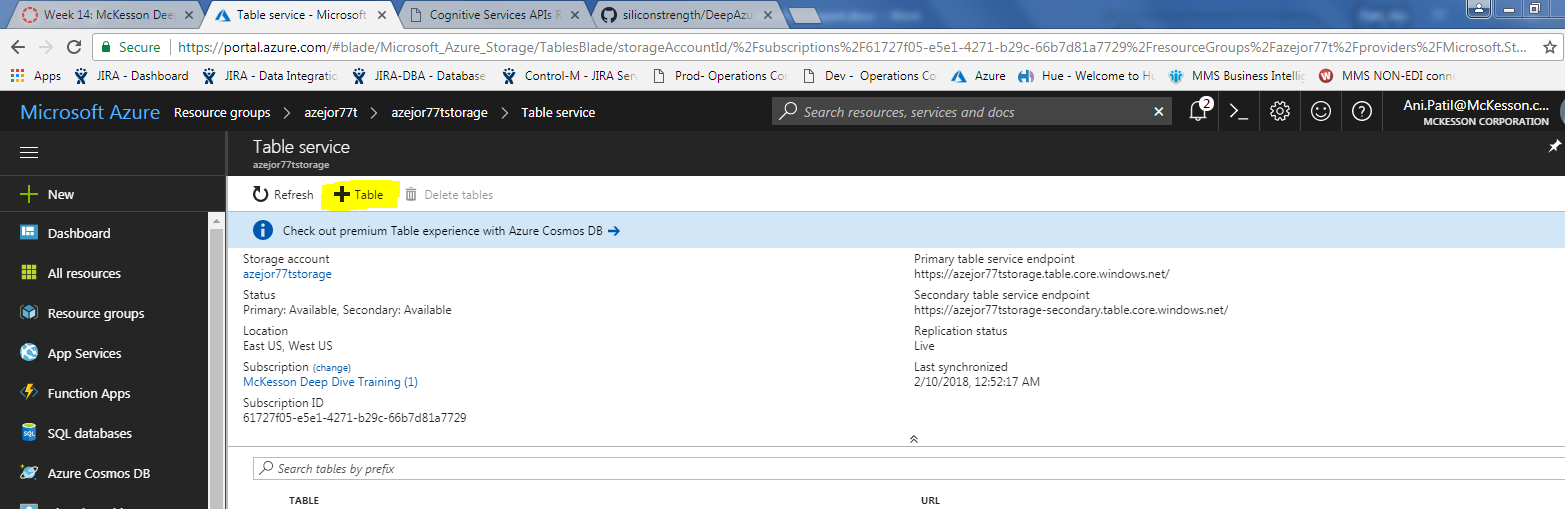
Create storage account



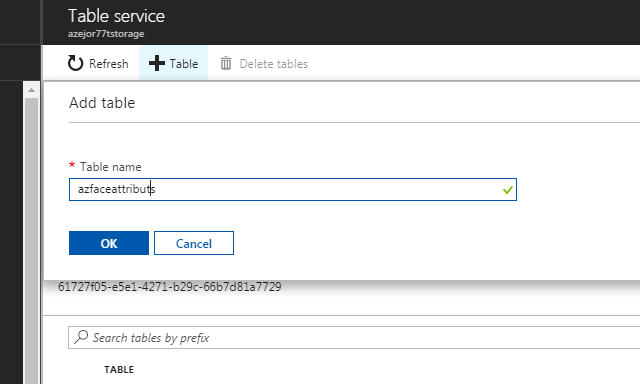
Once storage account is created, it look like below-



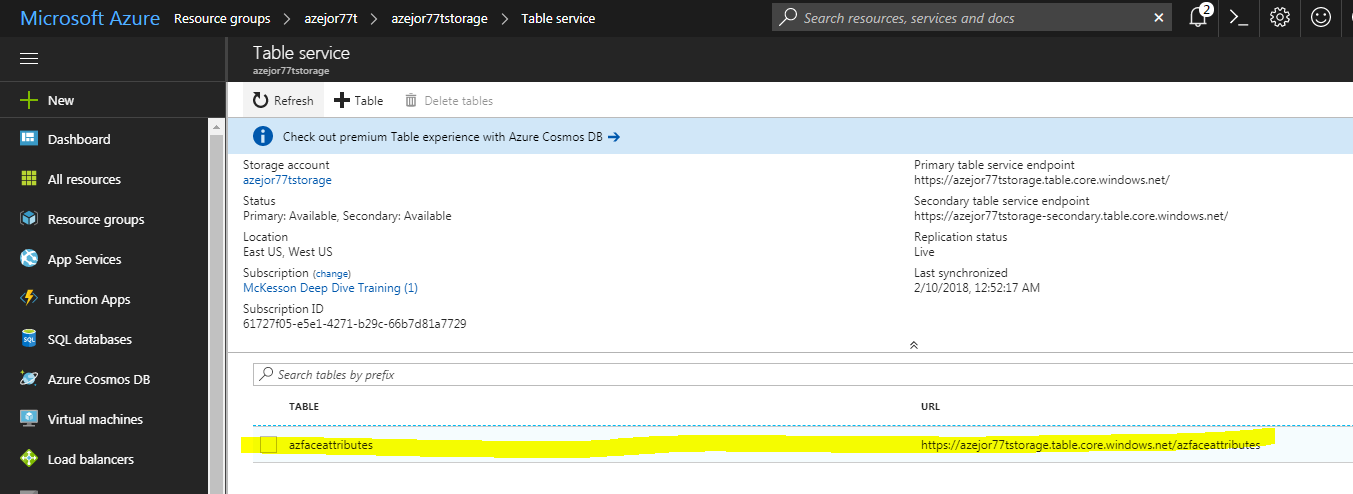
Select ‘Tables’ above and then click on + sign as below to create a new table



Give name of new table-



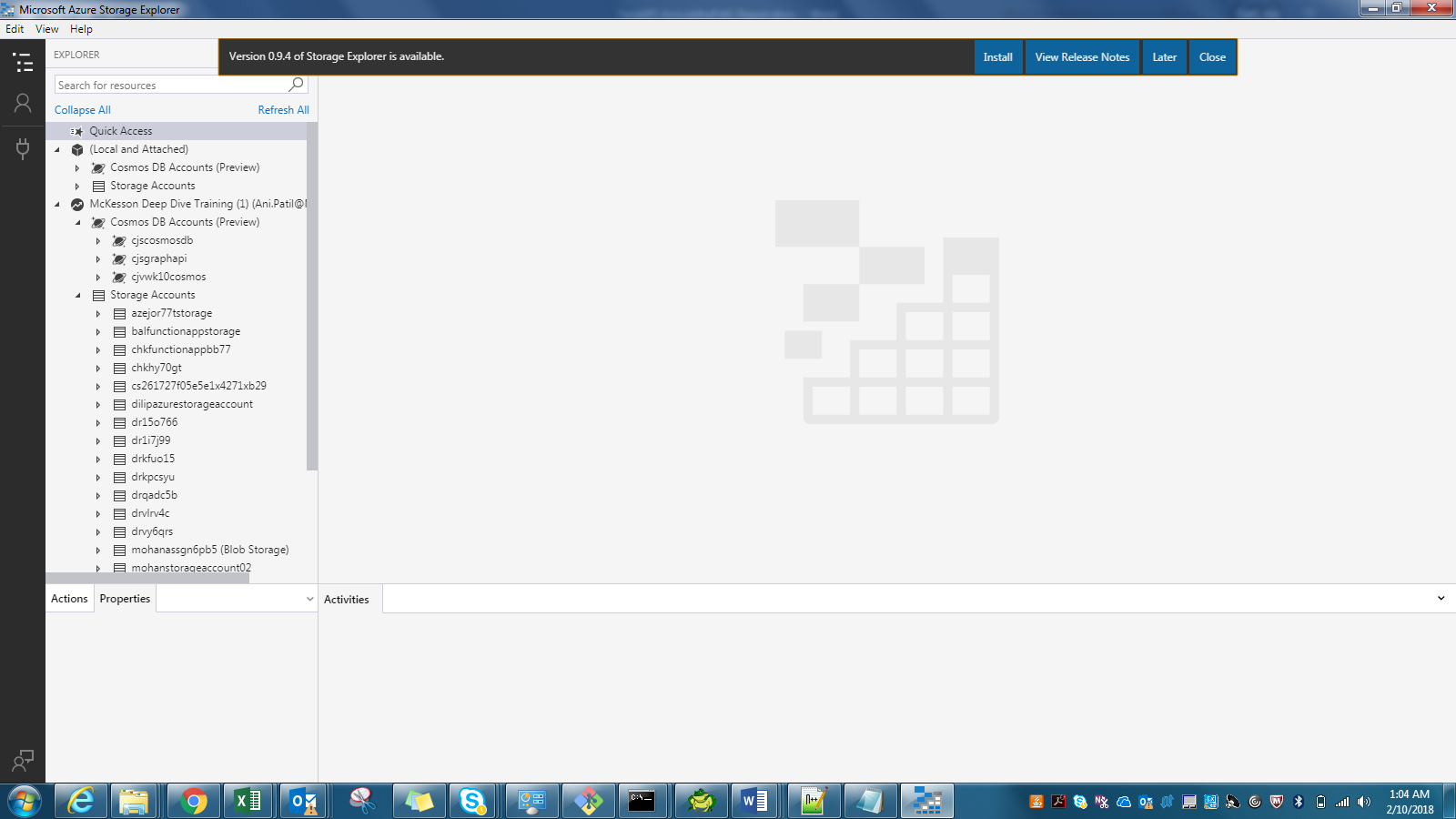
Once table created, it will look as below-



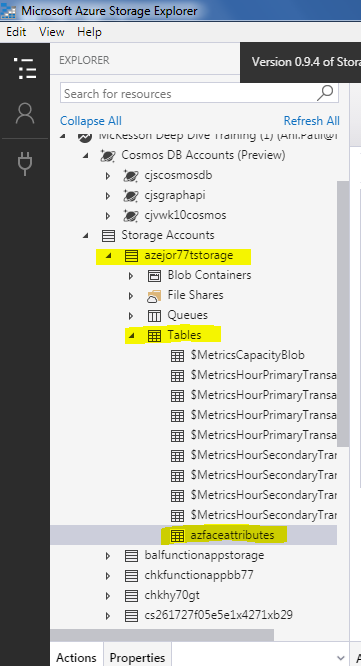
To access and manage this table, we will need to install – Microsoft Azure Storage Explorer. You can download it from-

<https://azure.microsoft.com/en-us/features/storage-explorer/>

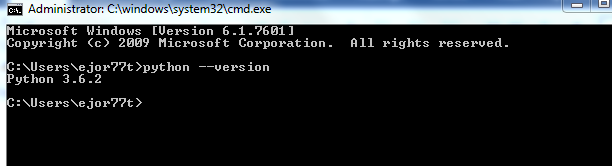
Microsoft Azure Storage Explorer, when first opened, will ask for your Azure login credentials to connect to Azure storage. It looks as below-



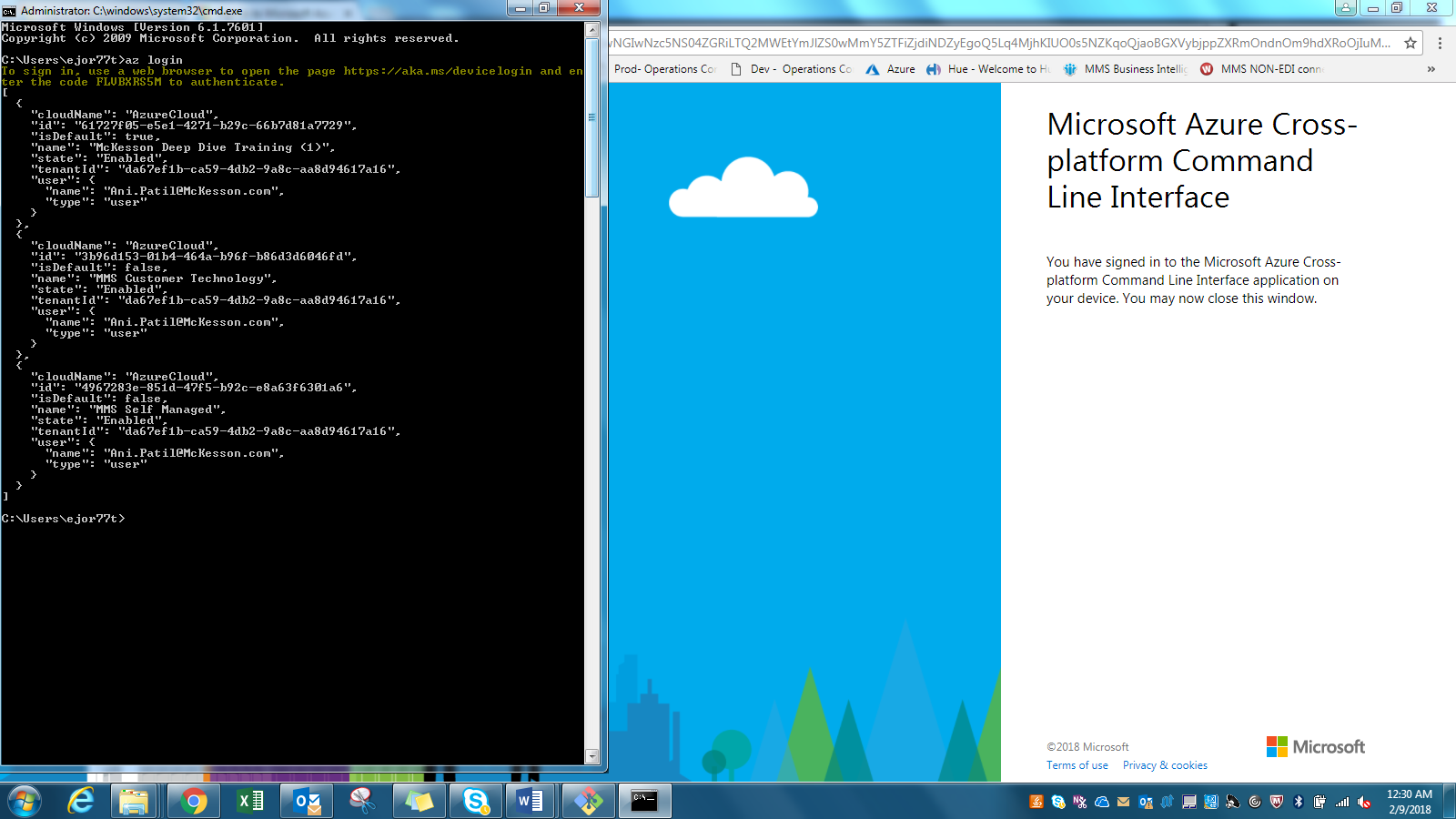
Once you click your storage Account, it will display respective tables inside it



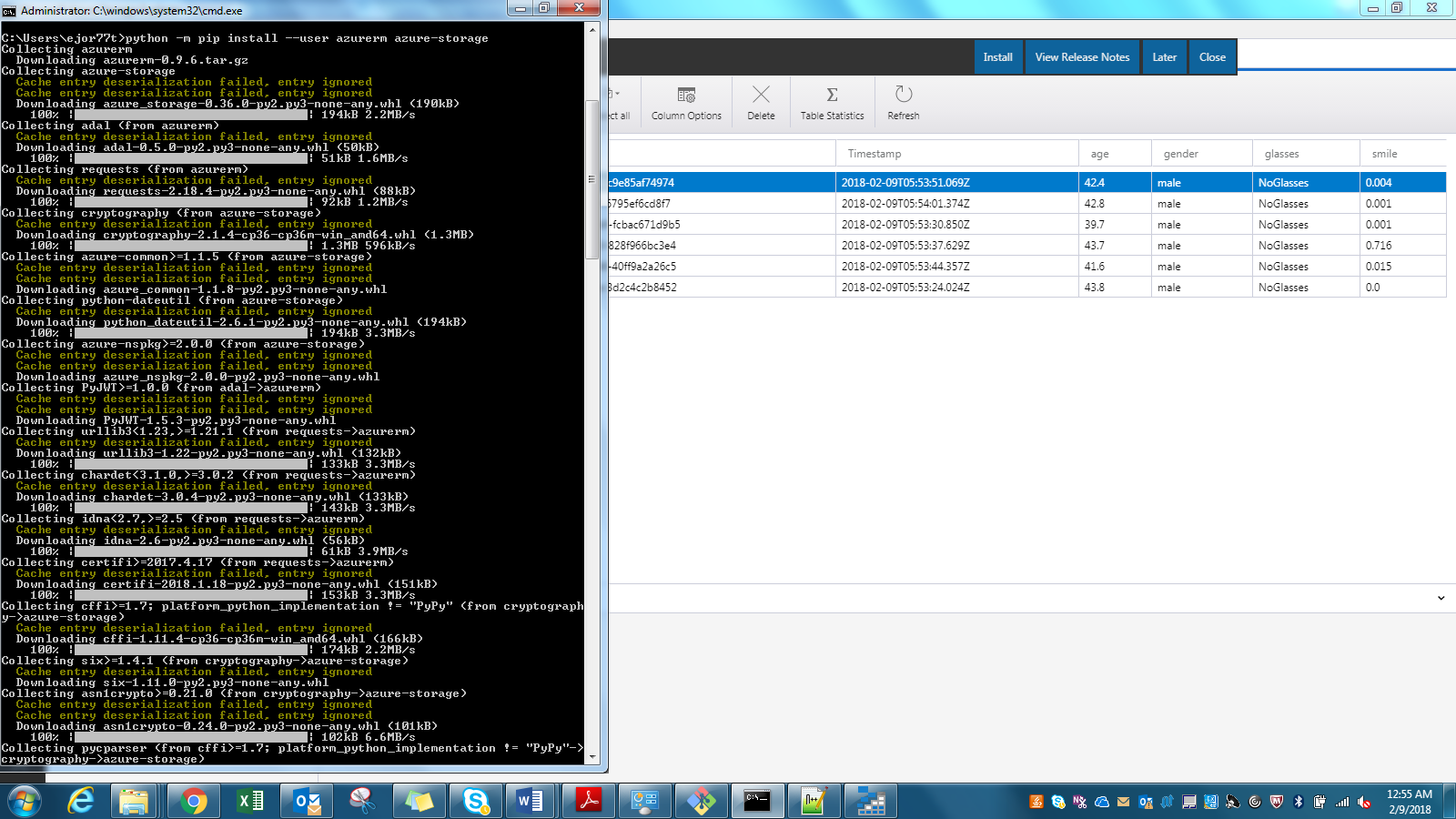
Then check that your machine has Python 3.6 installed (if not get it installed).



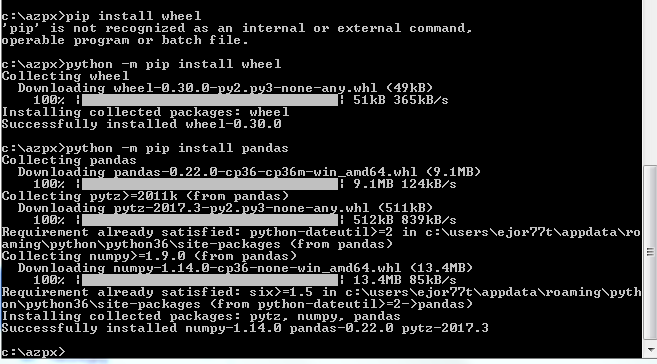
Now lets login to Azure using Azure CLI



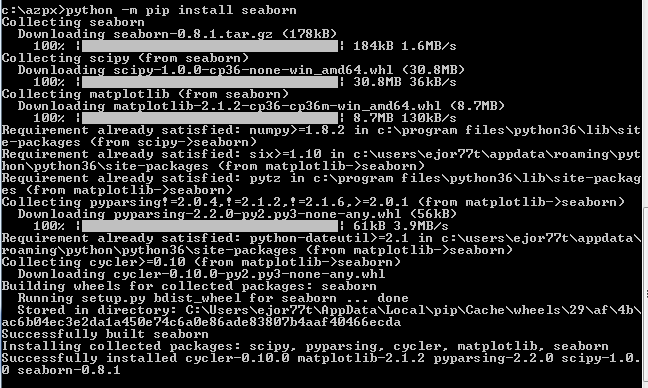
Install storage package needed for working with Azure storage table



Install wheel and pandas – This we will need later for visualization.



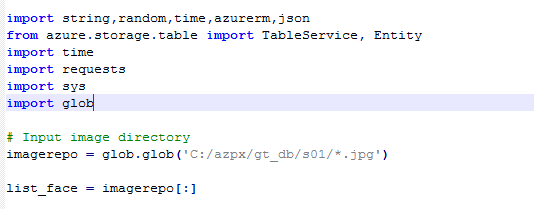
Install seaborn – needed for data visualization later



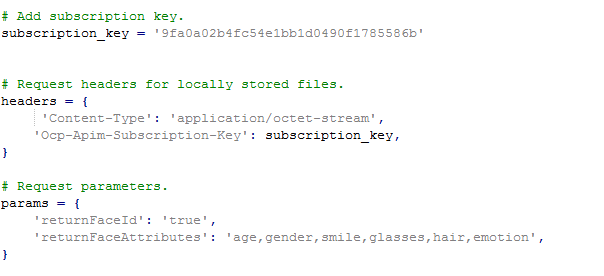
Now lets start writing code for Face API call and do some face detection and get face attributes.

Below is code snippet of my Face API program- azfaceapi.py

Setting up subset of input images pulled from above Face Database to be processed by Face API-



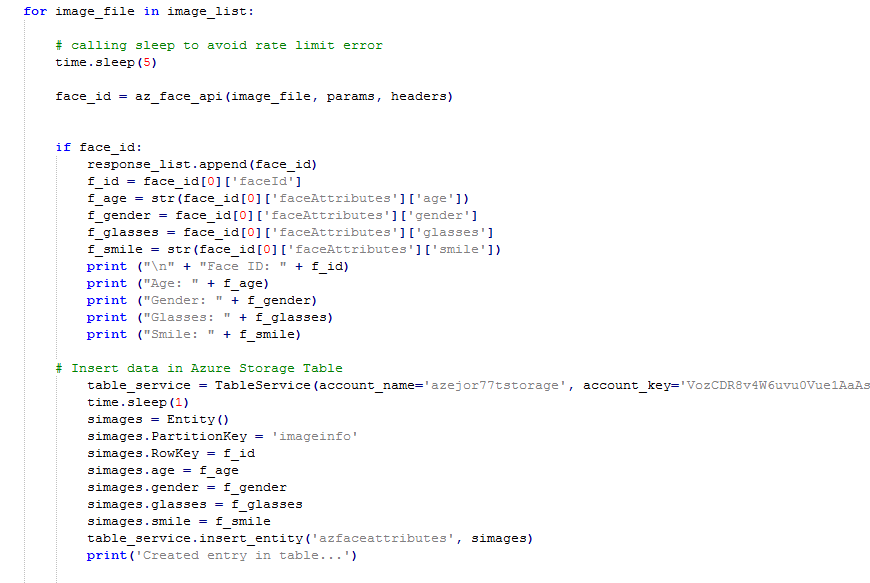
Set subscription key for face API and header and params for request



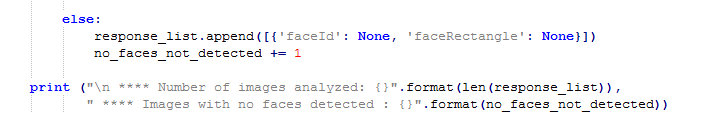
Next will call Face API for each of input image to analyze and collect face attributes



Once we get requested face attributes, result will be in JSON format. We will collect respective values and store them in Azure Storage table as below



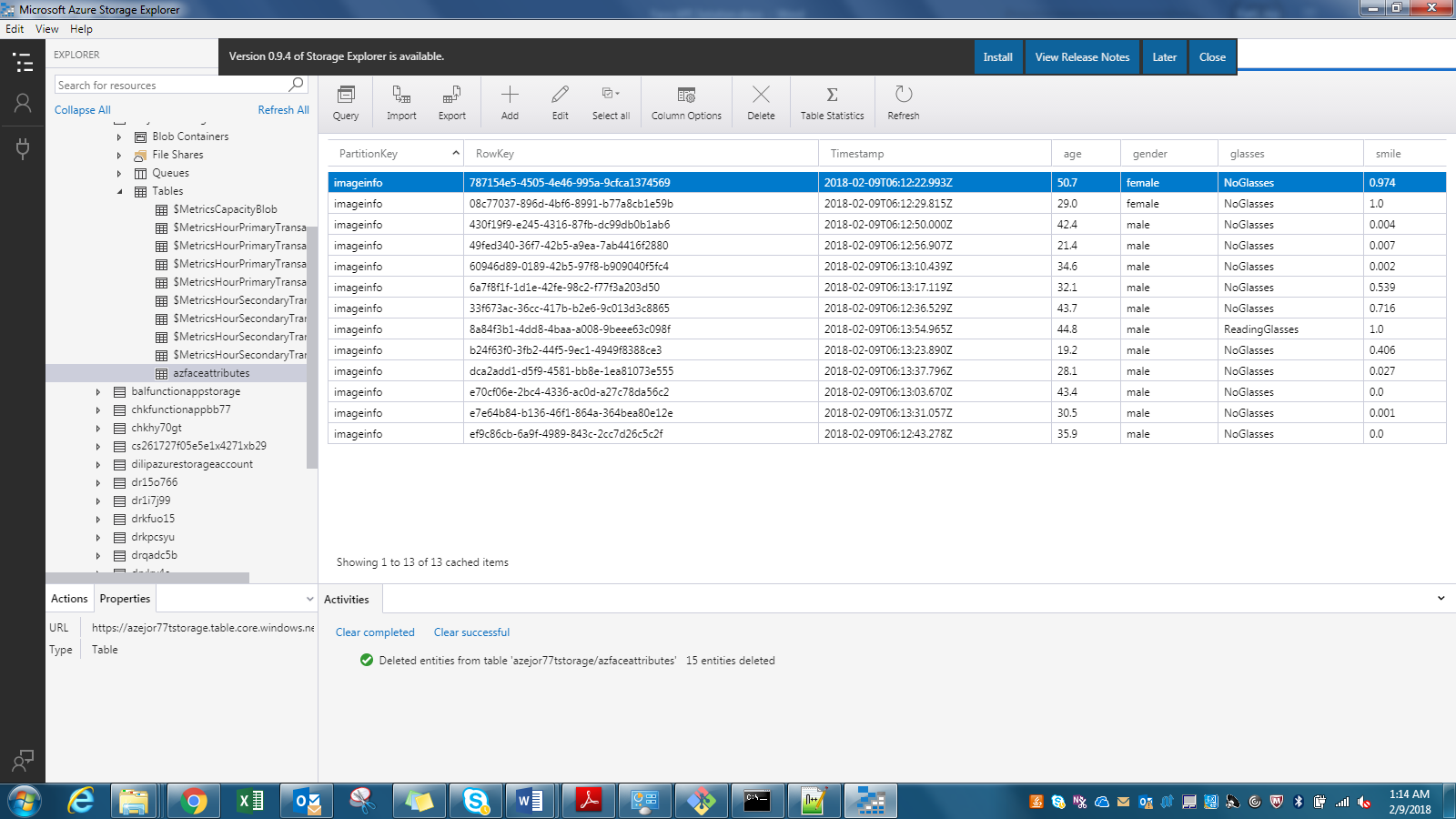
We will also be checking for images with no faces in it and will count such images as below



We can run program as below-



Output of Face Detection Analysis is stored in Azure Table Storage and when looked through Microsoft Azure Storage Explorer will look as below-



Now we will run some data visualization on output data in above storage table. Please extract this table data as csv file – azfaceattributes.csv

Now we will use Pandas to run visualization. I have created python program proj\_viz.py for same.

Python 3.6 on windows 7 machine was giving error on some of python pandas package installation so I ran visualization using Python 2.7 on Windows 10 machine.

In this python code, I have calculated some statistical figures covering group of faces analyzed. It pulls minimum, maximum, average, mean, median age and smile factor of group.

Visualization shows average age and average smiling face attribute for group of faces analyzed. I created Bar chart showing these averages.

Below is code snippet-

--------------

## Data visualization using Pandas on Face attribute data. Find average Age and average smiling face attribute of batch of input faces

## Run on Python 2.7

import pandas as pd

import seaborn as sns

from pandas.plotting import scatter\_matrix

import matplotlib.pyplot as plt

#Read azfaceattributes.csv file which is export of Face API attributes from Storage table

df = pd.read\_csv("azfaceattributes.csv")

#Drop non numeric columns and buid a Dataframe of remaining columns to work with

df1 = df.drop(['PartitionKey', 'RowKey','Timestamp','gender','glasses'], axis=1)

attributes=list(df1.columns)

#Use panda machinery to calculate basic statistics for all numerical columns, min, max, median, average and standard deviation

print "\n\*\*\*Applying min() : "

print df1.min()

print "\n\*\*\*Applying max() : "

print df1.max()

print "\n\*\*\*Applying median() : "

print df1.median()

print "\n\*\*\*Applying mean() : "

print df1.mean()

print "\n\*\*\*Applying std() : "

print df1.std()

#Present graphically average age and smiling attributes for attending group

df1.mean().plot(kind='bar')

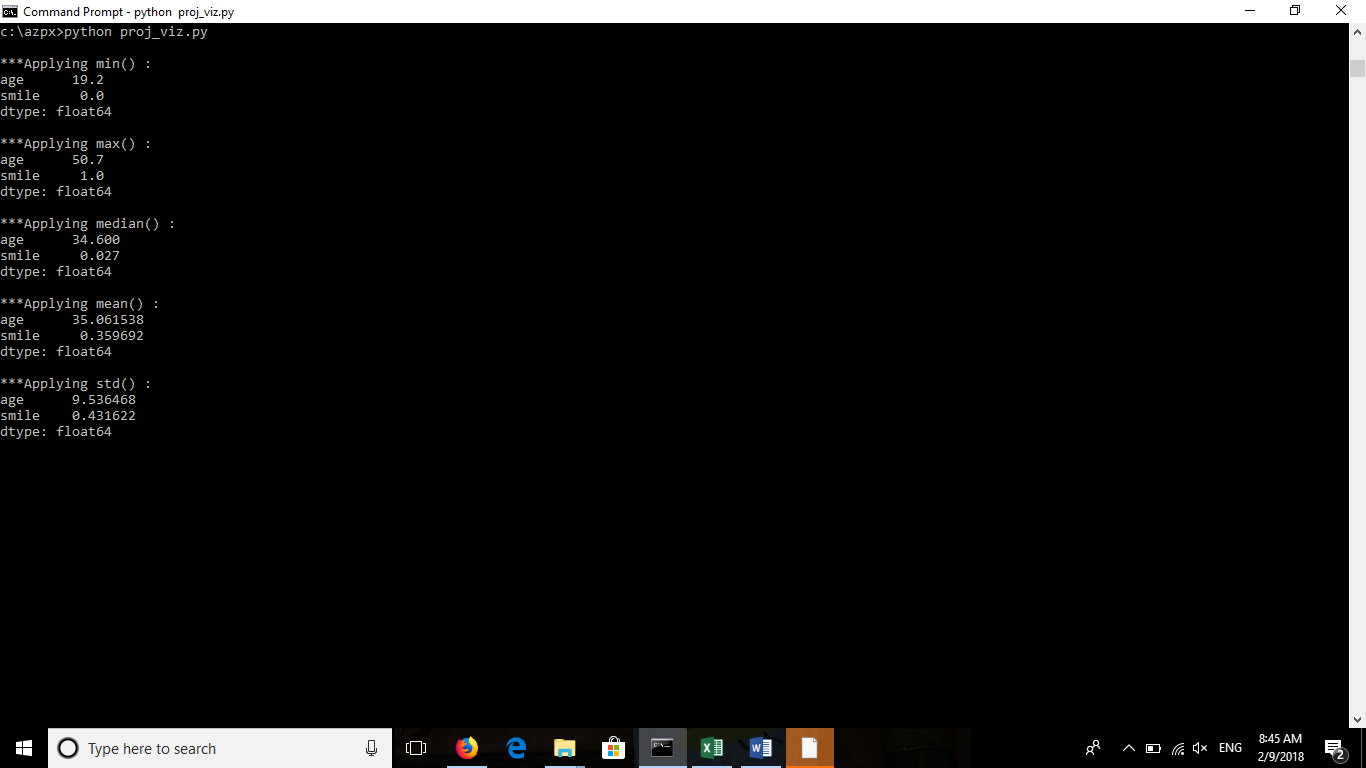
##scatter\_matrix(df1[attributes], figsize=(12,8))

plt.show()

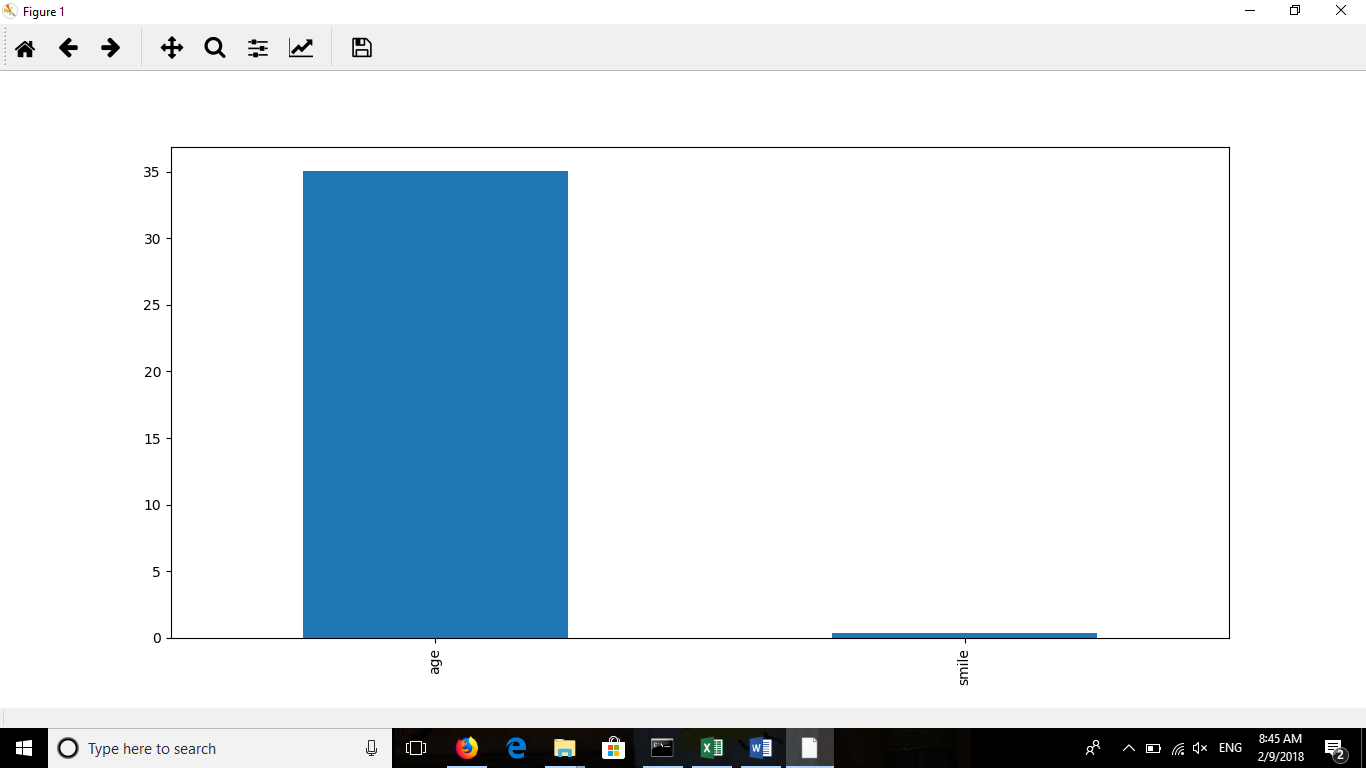
-------------

Above program will show data visualization as bar chart for average age and smile face attribute for all image faces for input data. Such data can be used for deriving social intelligence from public events.

Visualization looks as below-



Visualization is as below



You can see that average age of group is 34.6 and Smile factor of 0.35.

This can give us some insight on how specific group reacted to situation at that time.

**YouTube URLs :**

Short: <https://www.youtube.com/watch?v=l_3xt2chxns>

Long: <https://www.youtube.com/watch?v=7dBOHjJYPj4>

**Github :**

<https://github.com/siliconstrength/DeepAzureFinalProject>

**References:**

<https://azure.microsoft.com/en-us/services/cognitive-services/face/>

<http://www.anefian.com/research/face_reco.htm>

<https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview>