# CommOps Analytics -API Case study

## Requirement

The community operations team at uber uses Zendesk, a support ticketing system to effectively manage customer tasks. The main objective of this was to create an effective ticket tracking system for the Workforce Management system.

## Code Blocks

## TicketTracker.html

Front end login screen

**Fields**
Domain - The domain the user wants to monitor
Username - The user name of the user
Password - The secure password

(For the purpose of my analysis of performance the following credentials are shared)
Domain - wmt
Username - arunramsank@gmail.com
Password - Customerservice@4591

Passwords are case sensitive and should be kept secured.

## Retrieve.php

Middle tier operations screen

**Input (fed by the previous process)**
Domainname - wmt (For the purpose of my analysis)
Username - arunramsank@gmail.com
Password - Customerservice@4591

**Displays**

Table containing View ID, Titles and Counts
Bar chart containing the visualization of the counts for each Ticket ID ( Hover over each bar to see their ID and Counts more clearly)

**Operations**

Has session variables to retain session username and password. Logs out automatically after inactivity of 30 minutes for the purpose of security.

Each call to the Tracker.py python file is made from this file. Each call is made in a duration of 10 minutes.

## Tracker.py

**Input (fed by the previous process)**
Domainname - wmt (For the purpose of my analysis)
Username - arunramsank@gmail.com
Password - Customerservice@4591

Passwords are case sensitive and should be kept secured.

**Output**
Count.csv - Contains the excel file which gives the Id, Title and count of each of the views

**Purpose**
Contains the code block that effectively crawls the Zendesk website and obtains the counts for different views.

## Environment

The code was tested in a Linux server . The version of python used was Python Version 2.7

## Prerequisites

- Python needs to be installed
- Python statements given in the import block needs to be installed in the machine

## Process

The process designed for this approach was a simple 3 step process. Since the efficiency of the process was more important than the design a simple web application was used for the purpose of interaction. The following are the steps

1. Enter credentials and click on submit
2. Analyze tweet counts and visualization to get better intuition of the counts
3. Close the screen or click on back button on the browser to login from a different domain

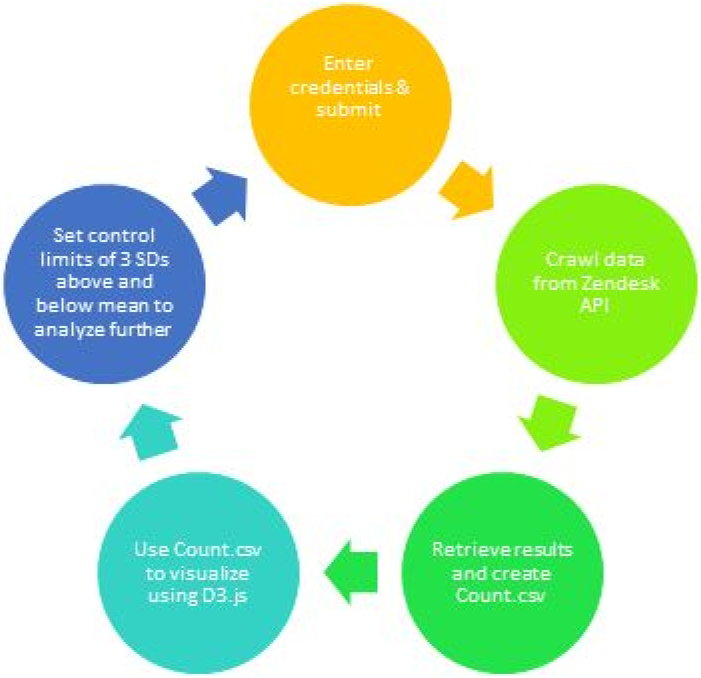
Enter credentials & submit
Crawl data from Zendesk API
Retrieve results and create Count.csv
Use Count.csv to visualize using D3.js
Set control limits of 3 SDs above and below mean to analyze further

Fig 1: Overview of the initial process flow

## Process Improvements and future plans

Since this was just a short challenge the scope for analyzing and improving was very little. Taking into account the sparse availability of resources there are numerous other problems that could be solved in future.

1. Time series modelling can be done to see amount of flow of tickets over a period of time , thereby enabling us to model and effectively analyze data
2. Control charts can be prepared to see the amount of tickets for a periodic time period. The rise of the amount of tickets of about three standard deviations above or below the mean means there is some anomaly in the data and they need to be investigated immediately.
3. Pareto charts and Gaant charts can also be used to effectively understand the root causes of ticket inflows.
4. Effective regression analysis can be done by analyzing the counts and other factors which could have a linear correlation
5. Effective clustering can be implemented to group similar agents together by using their ticket solving abilities thereby we could allocate our most effective Agents to the most appropriate tasks.
6. Creation of a windows application to create and monitor a dashboard can be something to look upon.

7. Effective visualization can also help people in intuitively understanding big data with just a quick look.
8. Problem records can be linked to Tickets and Tickets could be linked to the assignee. With this a directed graph relation can be established. A force directed graph is an intuitive visualization technique that could be used for this process.

## In progress

Currently the work is on creating windows and web applications. A head start on the windows application is in progress as below. A windows application would be the ideal in these situations since the ticket monitoring would involve continuous inflow and outflow of tickets and it makes sense to have a tool on the Desktop that could be referred and monitored periodically. The below is just a screenshot of the sample windows application that is in progress

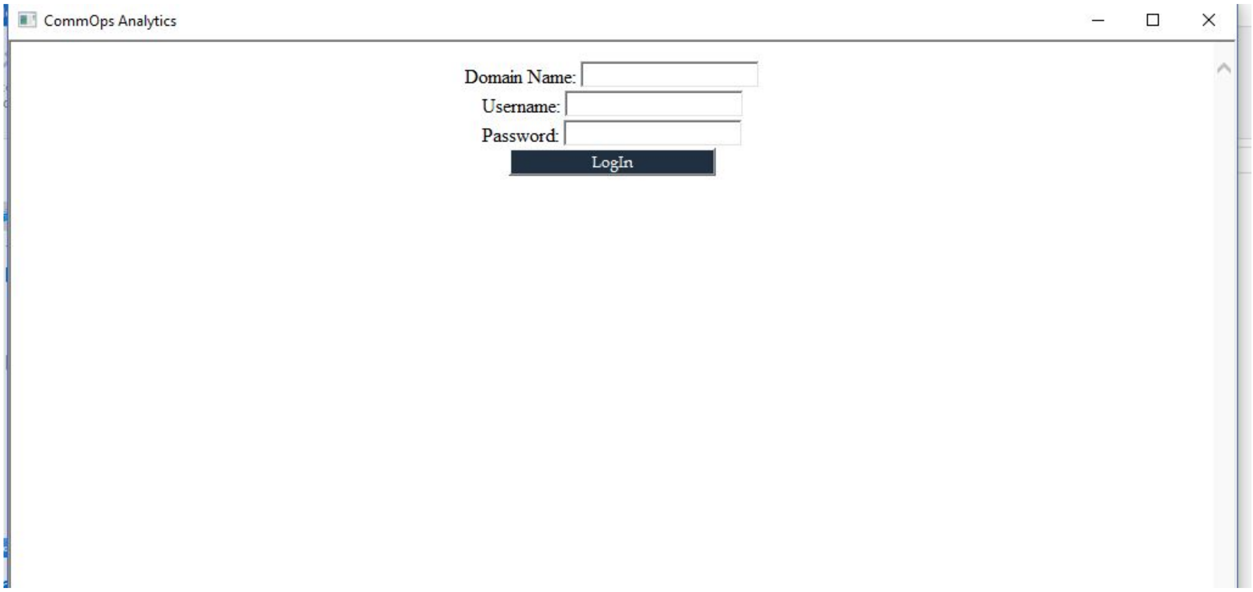

Fig 2 :A snapshot of the windows application

## Conclusion

The scope of this problem is wide ranged. There are lots of possibilities for process improvements and statistical analysis. When given with a good collection of data they can be monitored and intuitively analyzed. The simple system created above caters to a small set of needs within a group and can be expanded to cover wider horizons. We could create force directed graphs which could intuitively link multiple nodes together and help us analyze the flow much better.