# **CS-521 Project – Information structures with Python**

**Table of Contents**

[CS-521 Project – Information structures with Python 1](#_Toc3115121)

[**Introduction:** 1](#_Toc3115122)

[**Functionality:** 1](#_Toc3115123)

[**Installation instructions** 1](#_Toc3115124)

[**Outputs:** 2](#_Toc3115125)

[**Test cases** 3](#_Toc3115126)

**Author:** Ashwin Arunkumar

**Email-ID:** ashwinar@bu.edu

## **Introduction:**

This python project is a model for plotting graphs based on telemetry sensor data from a target device. The target device can be networking equipment like routers, switches, firewall, servers or even an application running on a cloud installed with various sensors. The target devices streams data to a client and stores the data in CSV format at the client. Data streaming from target devices to client is outside the scope of this project. This project assumes that the telemetry statistics data is available in table format at the client machine.

## **Functionality:**

This python code takes one input file which is telemetry data in CSV format. This data is further parsed using various python data structures such as dictionary, sets and lists at different levels. The keys in dictionary is mapped to its child node based on the raw data from CSV file. Later dictionary iteration is used in different forms along with set and list data structures. The final output is table with series of rows containing timestamp since epoch, interface name and counter value with tags and field keys for influxdb consumption. This table is further converted into a JSON file and the JSON file is written to influxdb, which is a time series database used for plotting graphs. In this code python uses influxdb API to write data to the database. Once the table is written to influxdb, any graphing tool that supports influxdb can be used to plot graph. In this case Grafana will be used to plot and visualize graph.

## **Installation instructions**

**Install influxdb using the below documentation:**

<https://www.influxdata.com/blog/getting-started-python-influxdb/>

Once installation is complete, start the influxdb from a terminal before executing the python code:

$ influxd -config /usr/local/etc/influxdb.conf

Check if the influxdb port is in listen mode based on your base operating system:

**For MAC:**

$ lsof -i :8086

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

**influxd 46194 ashwinak 29u IPv6 <snip>. 0t0 TCP \*:8086 (LISTEN)**

**Install Grafana using the below documentation:**

<http://docs.grafana.org/installation/>

## **Outputs:**

**Size of input file is 284MB.**

**CS521-Spring1-2019-Project ashwinak$ python main.py**

﻿Enter the raw telemetry CSV log file : cs521-stats.txt

Writing data to influxdb...

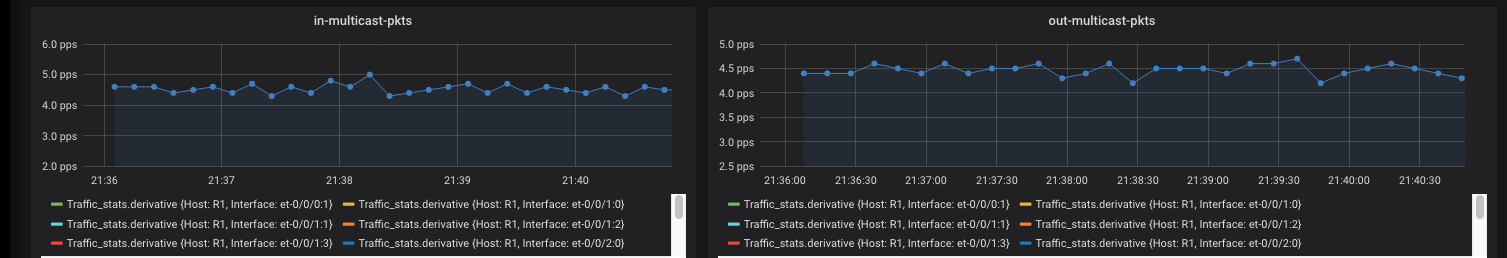
Writing data to influxdb Complete.

Time taken to write data to influxdb is, 666 seconds

Grafana is running on local machine and can be accessed via web browser (sample link).

<http://localhost:3000/d/J_9wdLjmk/cs521-dashboard?tab=queries&from=now-5m&to=now>





## **Test cases**

1. When influxdb port is not listening or influxdb is not installed, the following errors will be seen:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

%%%% Influxdb is not running,check readme.md for more details. %%%%

To run influxdb for python, follow this link

https://www.influxdata.com/blog/getting-started-python-influxdb/

After installation, run this command from a terminal before executing this code

$ influxd -config /usr/local/etc/influxdb.conf

1. When user interrupts in the middle of code execution:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

Enter the raw CSV log file: cs521-stats.txt

Writing data to influxdb...

^C

**User interrupted**

1. Time taken to write data to influx db is calculated on every successful run:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

Enter the raw telemetry CSV log file with 2 columns of format (key,value): cs521-stats.txt

Writing data to influxdb...

Writing data to influxdb Complete.

**Time taken to write data to influxdb is, 572 seconds**

1. When the input file does not exist:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

Enter the raw CSV log file: a.txt

**The file a.txt is not found from the location you are running this code**

1. When the input file is empty:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

Enter the raw CSV log file: a.txt

**The file a.txt is empty**

1. When undefined exception is seen by the code:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

Enter the raw CSV log file: a.txt

**Printing the exception for further debugging...**

**No columns to parse from file**

1. Catch all exception:

**CS521-Spring1-2019-Project ashwinak$ python main.py**

**General error!!Comment this piece of code to debug further**

1. When python2 is used to execute this code, the following errors will be seen:

**$python2 main.py**

%%%Usage : python3 main.py

%%%% Influxdb,Pandas or some other module is not installed ,check readme.md for more details. %%%%

To install and run influxdb for python, follow this link

https://www.influxdata.com/blog/getting-started-python-influxdb/

To install influxdb, use this command:

$ python3 -m pip install influxdb

To install pandas, use this command:

$ pip3 install pandas

1. When this code is run with python3 but on a new environment without any dependencies, the following errors will be seen:

$ **python3 main.py**

%%% Usage: python3 main.py

%%%% Influxdb,Pandas or some other module is not installed ,check readme.md for more details. %%%%

To install and run influxdb for python, follow this link

https://www.influxdata.com/blog/getting-started-python-influxdb/

To install influxdb, use this command:

$ python3 -m pip install influxdb

To install pandas, use this command:

$ pip3 install pandas

1. When the input CSV file has errors i.e. not in proper CSV format or the number of columns vary, the errors will be logged to a different file instead of stdout:

more bad\_lines.txt

b'Skipping line 4184: expected 2 fields, saw 3\nSkipping line 14121: expected 2 fields, saw 3\nSkipping line 27026: expected 2 fields, saw 3\nSkipping line 39931: expected 2 fields, saw 3\nSkipping line 52836: expected 2 fields, saw 3\nSkipping line 67536: expected 2 fields, saw 3\nSkipping line 80441: expected 2 fields, saw 3\nSkipping line 93346: expected 2 fields, saw 3\nSkipping line 106251: expected 2 fields,

<snip>