**Reference:**

1. https://john.soban.ski/refactor-python-to-influx-2.html

**Requirements:**

1. Get started with InfluxDB 2.0

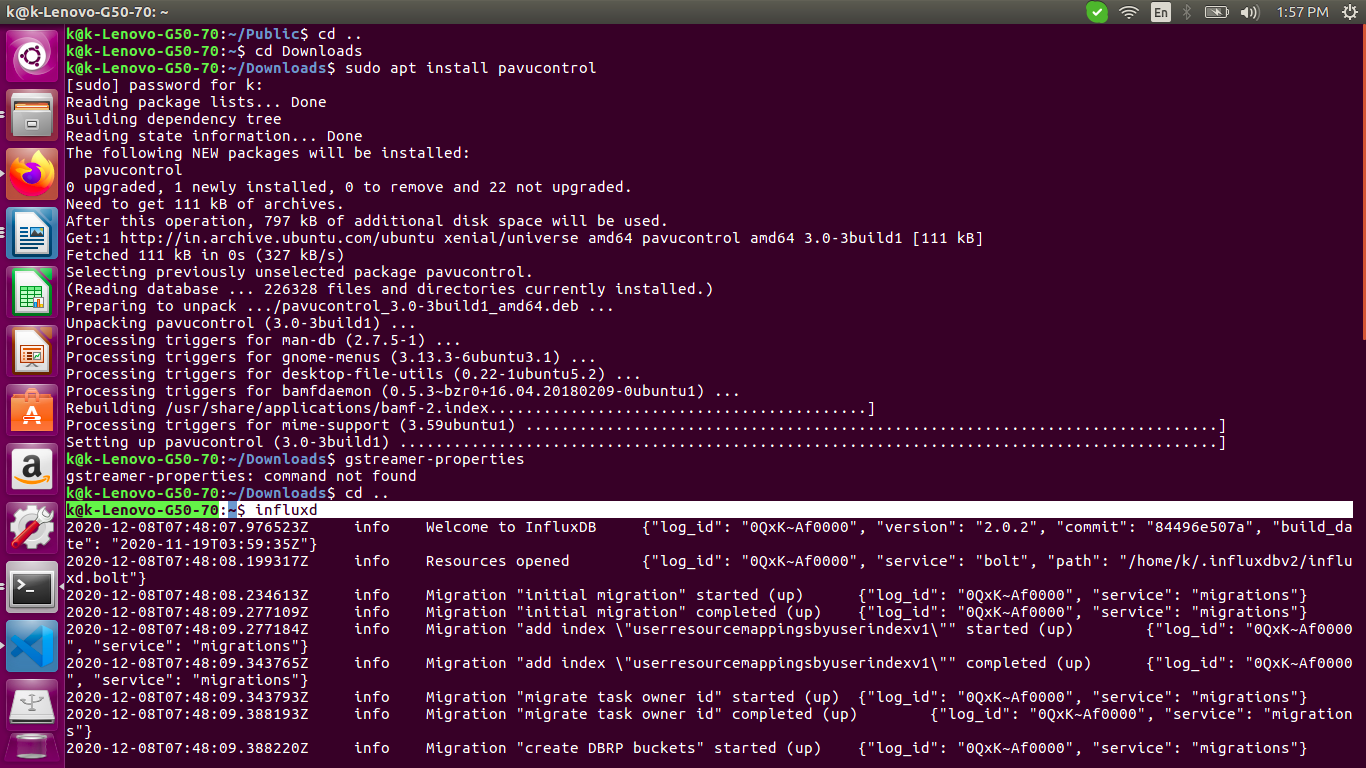
The InfluxDB 2.0 time series platform is purpose-built to collect, store, process and visualize metrics and events. Get started with InfluxDB OSS v2.0 by downloading InfluxDB, installing the necessary executables, and running the initial setup process.

If not installed, follow the link

https://docs.influxdata.com/influxdb/v2.0/get-started/

Start InfluxDB by running the influxd daemon:

k@k-Lenovo-G50-70:~$ **influxd**



2. Python installed

k@k-Lenovo-G50-70:~$ **python --version**

Python 2.7.12

k@k-Lenovo-G50-70:~$ **python3 --version**

Python 3.5.2

Steps:

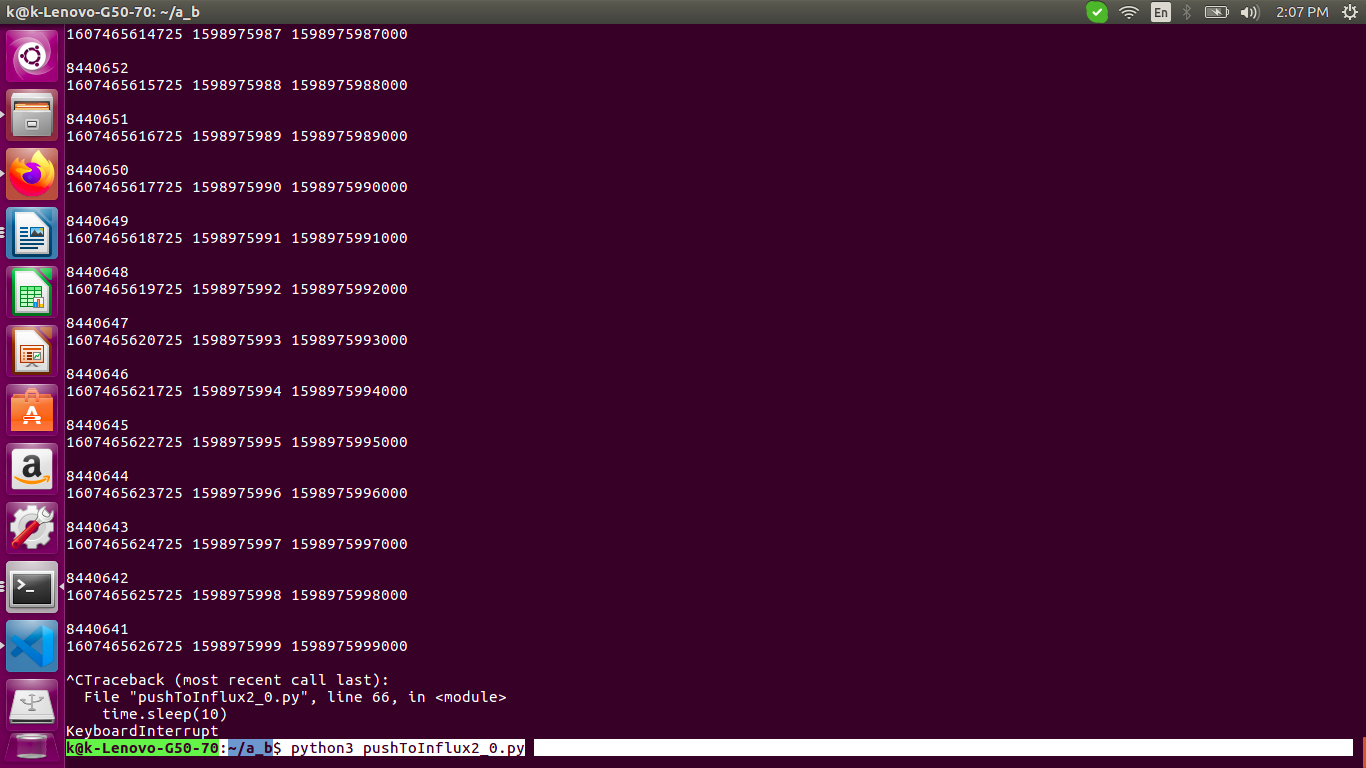
1. Replace the values of INFLUX\_TOKEN, ORG, BUCKET\_NAME and measurement\_name with the name of the table you need to create.

Also replace the csv path you need to upload the csv file at line:

with open('/home/k/Downloads/influxData/data\_0\_20200901.csv') as csv\_file:

2. Run the below program:

k@k-Lenovo-G50-70:~/a\_b$ **python3 pushToInflux2\_0.py**



/home/k/a\_b/pushToInflux2\_0.py

Code:

#!/usr/bin/python

import requests

import uuid

import random

import time

import sys

import csv

import json

INFLUX\_TOKEN='qCAYOyvOErIP\_KaJssk\_neFar-o7PdvHL64eWYCD\_ofywR\_J3iubktdB58A3TE-6sM7C61Gt8qOUPvc4t0WVBg=='

ORG="asz"

INFLUX\_CLOUD\_URL='localhost'

BUCKET\_NAME='b'

# Be sure to set precision to ms, not s

QUERY\_URI='http://{}:8086/api/v2/write?org={}&bucket={}&precision=ms'.format(INFLUX\_CLOUD\_URL,ORG,BUCKET\_NAME)

headers = {}

headers['Authorization'] = 'Token {}'.format(INFLUX\_TOKEN)

measurement\_name = 'data\_0\_20200901'

# Increase the points, 2, 10 etc.

number\_of\_points = 1000

batch\_size = 1000

data\_end\_time = int(time.time() \* 1) #milliseconds

id\_tags = []

for i in range(100):

id\_tags.append(str(uuid.uuid4()))

data = []

current\_point\_time = data\_end\_time

with open('/home/k/Downloads/influxData/data\_0\_20200901.csv') as csv\_file:

csv\_reader = csv.reader(csv\_file, delimiter=',')

print('Processed')

\_data\_end\_time = int(time.time() \* 1000) - (100 \* 1 \* 1000)

for row in csv\_reader:

\_row = 0

current\_point\_time = current\_point\_time - 1000

\_data\_end\_time = \_data\_end\_time + (1 \* 1000)

if row[0] == "TIMESTAMP":

pass

else:

\_add = int(time.time()) - int(row[0])

#\_row = int((int(row[0]) + 5847435 + 952068) \* 1000)

\_row = int((int(row[0])) \* 1000)

print(\_add)

print(\_data\_end\_time, row[0],\_row, '\n')

data.append("{measurement},location={location} POWER\_A={POWER\_A},POWER\_B={POWER\_B},POWER\_C={POWER\_C} {timestamp}"

.format(measurement=measurement\_name, location="reservoir", POWER\_A=row[2], POWER\_B=row[3], POWER\_C=row[4], timestamp=\_row))#timestamp=row[0]))......(data\_end\_time + 1000)

count = 0

if \_\_name\_\_ == '\_\_main\_\_':

# Check to see if number of points factors into batch size

count = 0

if ( number\_of\_points % batch\_size != 0 ):

raise SystemExit( 'Number of points must be divisible by batch size' )

# Newline delimit the data

for batch in range(0, len(data), batch\_size):

time.sleep(10)

current\_batch = '\n'.join( data[batch:batch + batch\_size] )

print(current\_batch)

r = requests.post(QUERY\_URI, data=current\_batch, headers=headers)

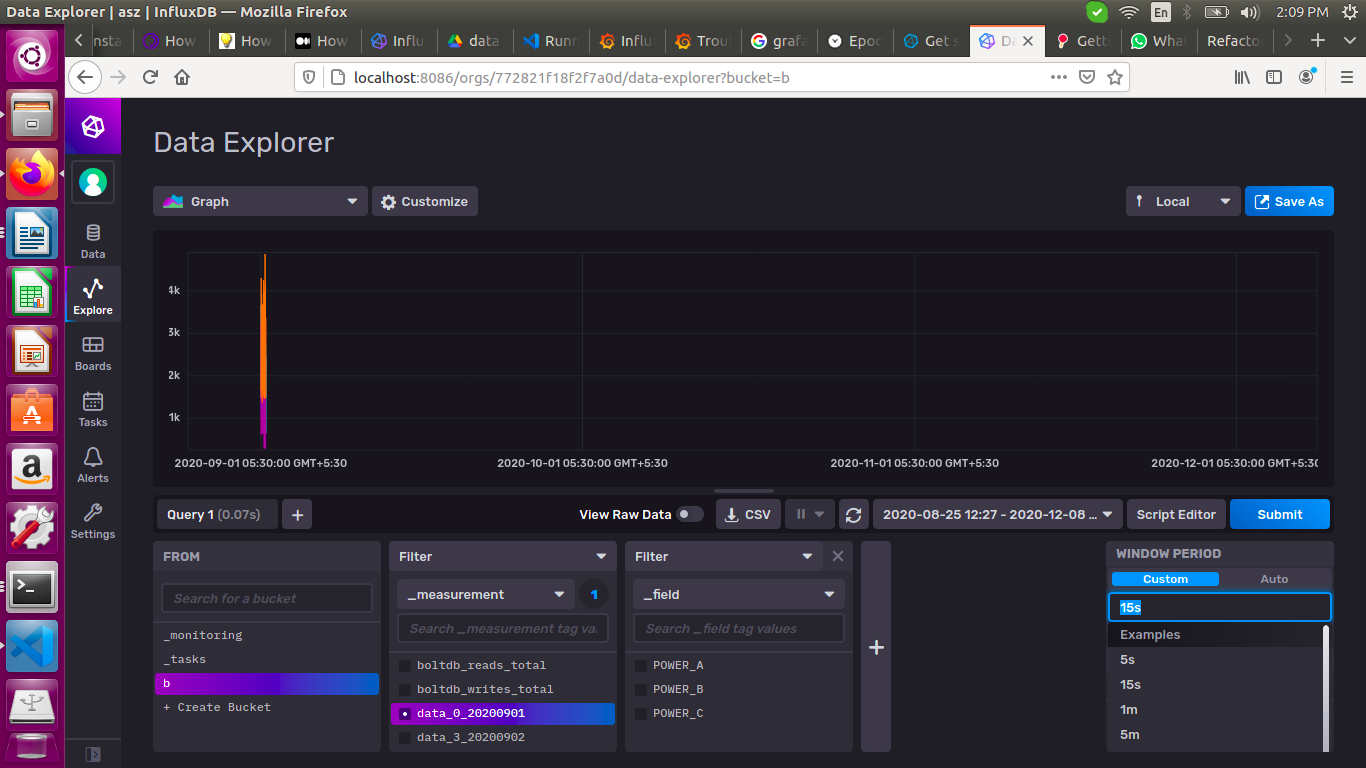
count = count + 1

print(r.status\_code, count, data[count])

3. In the InfluxDB2.0 screen page at:

<http://localhost:8086/orgs/772821f18f2f7a0d/data-explorer?bucket=b>

Under the Explore option



We need to set the WINDOW PERIOD as 15s / 1m to see more points on the dashboard.