**Install grafana and influxdb**

**Installing grafana on windows:**

Check whether the gcc is present or not by running “g++ -v” or

Install TDM-gcc if it is not present in your pc,you can download it form

<http://tdm-gcc.tdragon.net/download> go for 64 bit

here is link showing steps to install it <https://xcodelovers.wordpress.com/2011/02/01/tutorial-installing-tdm-gcc-windows/>

(note there are many ways ways to install gcc (ex: using minGw or cygwin ) this is just one of the way)

1).Open [h**ttps://grafana.com/grafana/download?platform=windows**](https://grafana.com/grafana/download?platform=windows)

Here click on the link given in “Standalone Windows Binaries(64 Bit)” to download latest version.

2). extract it

3). now go to “conf” folder, copy the “sample” file and paste it there and change it to “custom”.

4). open “custom” look for “http\_port = 8080” if this is commented then uncomment it. (“**;**” this is comment symbol)

5). go to bin folder run the “grafana-server.exe”.

6). Run <http://localhost:8080> on your browser.

7). Enter username : admin and password : admin.

**Installing influxdb on windows:**

1). Download influxdb from <https://portal.influxdata.com/downloads> in there click on influxdb and go for windows binaries.

2). Extract it

3). Open “influxdb.conf”

4). Now in the [meta] look for dir = "/var/lib/influxdb/meta" change it to a dir where you want to save data.(eg: dir = "C:/Users/xyz/influxdb/meta").similarly in [data] change dir = "var/lib/influxdb/data" (eg: dir = "C:/Users/xyz/influxdb/data”) and change wal-dir = "/var/lib/influxdb/wal" (eg: dir = "C:/Users/xyz/influxdb/wal).

5) further look for [http] and uncomment this lines if they are commented

enabled = true

bind-address = ":8086"

auth-enabled = false

realm = "InfluxDB"

log-enabled = true

6). Now run the “influxd.exe” and you can write queries in “influx.exe”. Default influxdb is running on 8086 port. You can also open it in browser at <http://localhost:8086>

7) user : root and password :root

**Adding influxdb database to grafana with :**

I am adding an example of python code to generate timeseries data and add it to influxdb. Copy paste it and save as .py

# -\*- coding: utf-8 -\*-

"""Tutorial on using the InfluxDB client."""

import argparse

import time

from influxdb import InfluxDBClient

def main(host='localhost', port=8086):

"""Instantiate a connection to the InfluxDB."""

user = 'root'

password = 'root'

dbname = 'example'

dbuser = ‘toor’

dbuser\_password = 'toor’

query = 'select Float\_value from cpu\_load\_short;'

client = InfluxDBClient(host, port, user, password, dbname)

print("Drop database: " + dbname)

client.drop\_database(dbname)

print("Create database: " + dbname)

client.create\_database(dbname)

print("Create a retention policy")

client.create\_retention\_policy('awesome\_policy', '3d', 3, default=True)

b,z=0.0,0.0

print("Switch user: " + dbuser)

client.switch\_user(dbuser, dbuser\_password)

for i in range(1,23):

if(i in range(0,5)):

z+=i

elif(i in range(5,10)):

z-=i

else:

z+=i

for j in range(0,59):

if(j in range(0,20)):

b+=j

elif(j in range(20,40)):

b-=j

else:

b+=j

if(z!=0 and z>1):

z=b/z

json\_body = [

{

"measurement": "cpu\_load\_short",

"tags": {

"host": "server01",

"region": "us-west"

},

"time": "2018-07-28T"+str(i)+":"+str(j)+":00Z", #here you have to add current

"fields": { #date instead of 2028-07-28

"Float\_value": z,

"Int\_value": 3,

"String\_value": "Text",

"Bool\_value": True

}

}

]

time.sleep(5)

print("Write points: {0}".format(json\_body))

client.write\_points(json\_body)

'''client = InfluxDBClient(host, port, user, password, dbname)

print("Create database: " + dbname)

client.create\_database(dbname)

print("Create a retention policy")

client.create\_retention\_policy('awesome\_policy', '3d', 3, default=True)

print("Switch user: " + dbuser)

client.switch\_user(dbuser, dbuser\_password)

print("Write points: {0}".format(json\_body))

client.write\_points(json\_body)'''

print("Querying data: " + query)

result = client.query(query)

print("Result: {0}".format(result))

print("Result: {0}".format(result))

print("Switch user: " + user)

client.switch\_user(user, password)

'''print("Drop database: " + dbname)

client.drop\_database(dbname)

'''

def parse\_args():

"""Parse the args."""

parser = argparse.ArgumentParser(

description='example code to play with InfluxDB')

parser.add\_argument('--host', type=str, required=False,

default='localhost',

help='hostname of InfluxDB http API')

parser.add\_argument('--port', type=int, required=False, default=8086,

help='port of InfluxDB http API')

return parser.parse\_args()

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

args = parse\_args()

main(host=args.host, port=args.port)

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

1). Start influxd and Run this code, it will create a database name “example” .

2). The python script will generate the json format data and it to the influx.

3). Now run grafana in browser.

4). You will see a “setting symbol” go there click on “datasources”, there you will see a “add data source” button click on it.

5). Give a any name you want,in “type” select “influxdb”, in url write <http://localhost:8086> ,select the “basic auth” write in “user: toor”, ” password : toor” .

6). In “database” write “example” and “user :root “, ”password :toor” i.e. the influxdb database that you have created using python code.

7)Save this.

8). Click on “plus” symbol select “dashboard”, select “graph” (here you can go for whichever dashboard tye you want, “graph” is just for this example )

9). Click on that empty dashboard and press “e” to edit it.(here you can write or create any query that you want to perform )

10). In datasources select “example” (here you can select any data source you want)

11). To write a query select a three lined button and click on “toggel edit mode”.

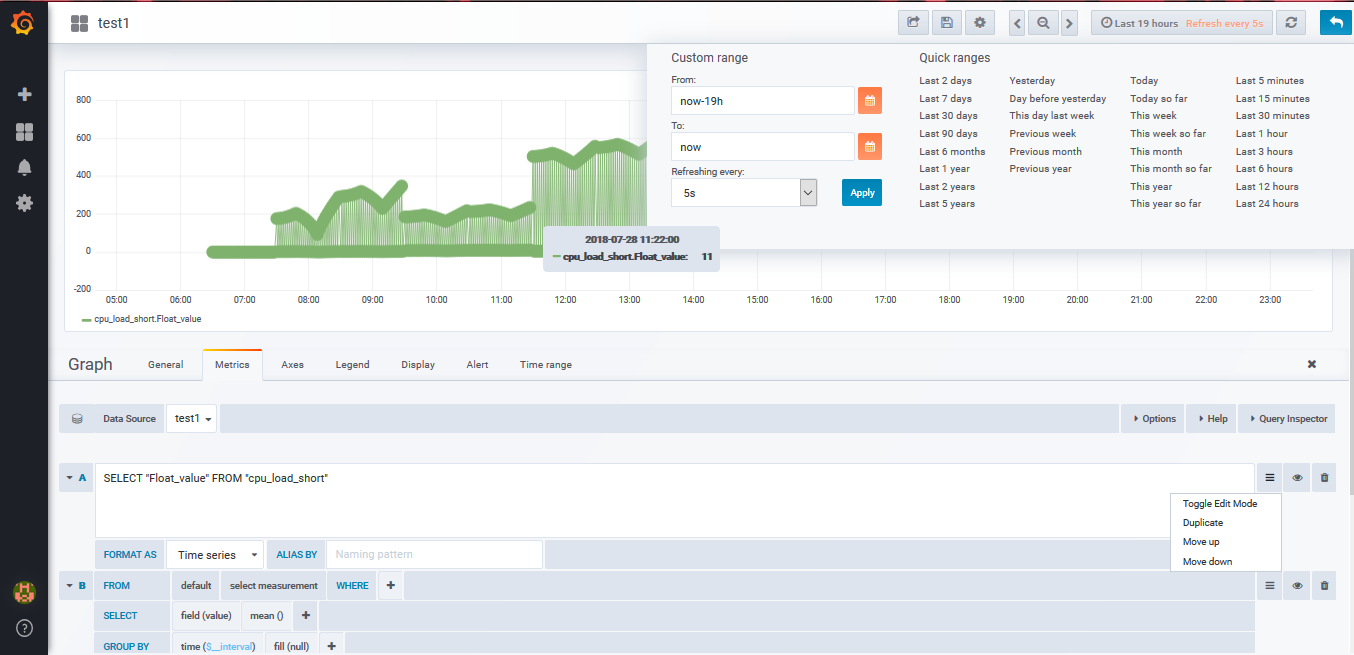
12). Write SELECT "Float\_value" FROM "cpu\_load\_short"

13). On top of right hand side you will that time is displayed there this for to show the data from which time to which, in that set “from : now-19h” , “to : now” and refresh rate as 5s.

14). Go to the display option mark write to lines and points.

15). Now you will be able to see that the points are generating on the graph at every five seconds.( if the points are not displayed check whether the python code is running ,if it is not then run it again).

Here is the screenshot of completely running this this example..

****

**Adding users:**

16) you can also add user with different level of permission as viewer,admin or editor in users in setting, you can send invite to do so open custom.ini in conf folder in grafana, search for [smtp]

**[smtp]**

**enabled = true**

**host = localhost:25**

**user = abc (uncomment all this lines and set it like as shown,you can**

**password = abc also set working mail address here)**

**cert\_file =**

**key\_file =**

**skip\_verify = false**

**from\_address = admin@grafana.localhost**

**from\_name = Grafana**

17). Now run grafana and go for to adduser and sent invite,look in pending invite and click on “copy link” to open for that user.

18). Paste it in browser you have to create the password and login.

19). You will be able to see only thing based on the role you have provided

**Sending static snapshot:**.

20). To send a static snapshot search for share in a dashboard which you want to share( it will be on top section).

21). There you will see to create snapshot option go on to that, here there will be two choices to create locally or to publish snapshot, go for them they will display a link copy it and open in any browser or you can send it to anyone. ( in static snapshot the live data will not be shown)

**Adding extra plugins for database and dashboard:**

To do so open plugins in grafana search for required plugins click on it, it will show an command, open the command prompt go to the directory of grafana in it go to bin and run that command. Now restart the grafana you will be able to see the changes.