Project Documentation of

“POC of time series data viz application using influxdb/timescaledb and grafana”

Required Dependencies:

**OS**: Windows/Linux (We are using windows here)

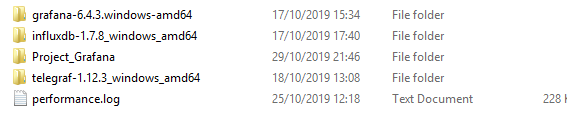
**Software/application**:

* InfluxDB v1.7.8 (Required for Time Series Data Strorage)
  + Download link (Win 64): <https://dl.influxdata.com/influxdb/releases/influxdb-1.7.8_windows_amd64.zip>
* Telegraf v1.12.4 (Data collection daemon for InfluxDB)
  + Download link (Win 64): <https://dl.influxdata.com/telegraf/releases/telegraf-1.12.4_windows_amd64.zip>
* Grafana v6.4.3 (Data visualizations/ dashboard)
  + Download link (Win 64): <https://dl.grafana.com/oss/release/grafana-6.4.3.windows-amd64.zip>
* Python v3.8.0 (Mock log generation)
  + Download link (Win 64): <https://www.python.org/ftp/python/3.8.0/python-3.8.0.exe>

As we are testing locally so portable (zipped) are used here so that it does not affect any other installed application or service, and can be easily terminated by closing the application window (except python we need it to install on the test machine)

**Installation:**

After all the dependencies are downloaded, unzip them in their designated folder/directory. I copied those into a single folder for convenience.



Check installation of python by opening command prompt and tying the following:

python –version

This will return the version of the python as following:

Python 3.8.0

**Setting up the environment:**

* Open the telegraf folder, you will find the following files. 
* We need to create our own Config file to initiate the telegarf process.
* Create a new text document and rename it to telegraf\_test.conf
* Open the telegraf\_test.conf file and add the following configuration

[[inputs.logparser]]

  ## files to tail.

  files = ["test\_run\_id.log"]

  ## Read file from beginning.

  from\_beginning = true

  ## Override the default measurement name, which would be "logparser\_grok"

  name\_override = "test\_metric"

  ## For parsing logstash-style "grok" patterns:

  [inputs.logparser.grok]

    #patterns = ["%{COMBINED\_LOG\_FORMAT}"]

    patterns = ["%{CUSTOM\_LOG}"]

    custom\_patterns = '''

      CUSTOM\_LOG %{TIMESTAMP\_ISO8601:timestamp:ts-"2019-10-16 17:01:52.905"}  .\*- %{WORD:status} .\*action:%{GREEDYDATA:action}(.\*)] .\*\(%{NUMBER:run\_dur:float} seconds\) .\*Run\_ID : %{GREEDYDATA:run\_id}

    '''

[[outputs.influxdb]]

  ## The full HTTP or UDP endpoint URL for your InfluxDB instance.

  urls = ["http://localhost:8086"] # required

  ## The target database for metrics (telegraf will create it if not exists).

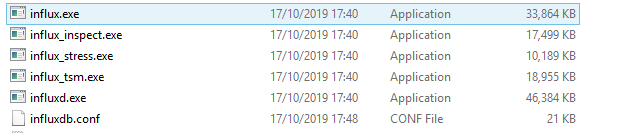
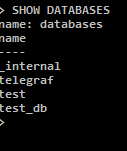
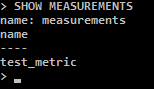
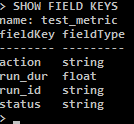
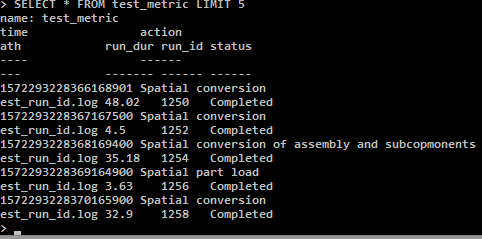
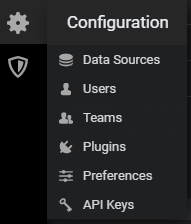
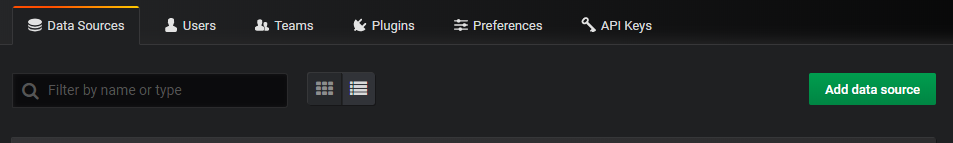
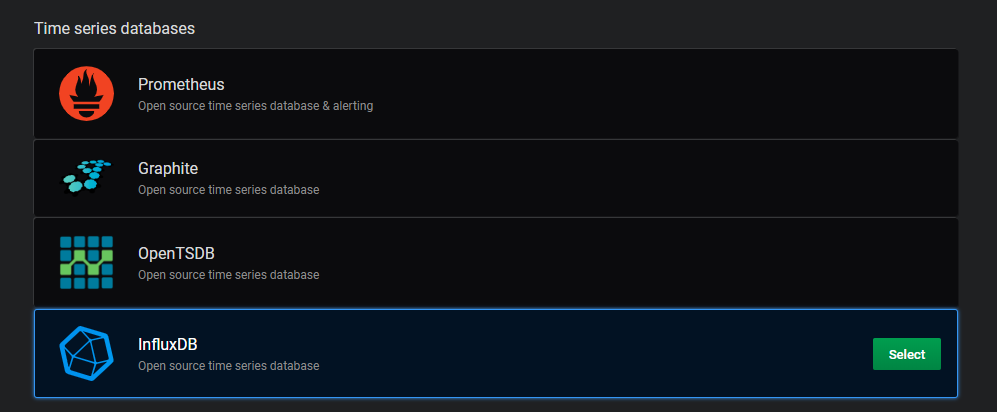
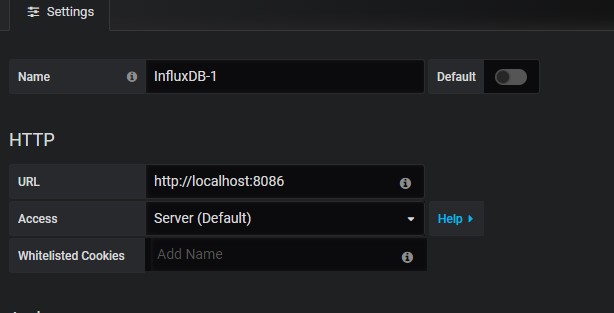
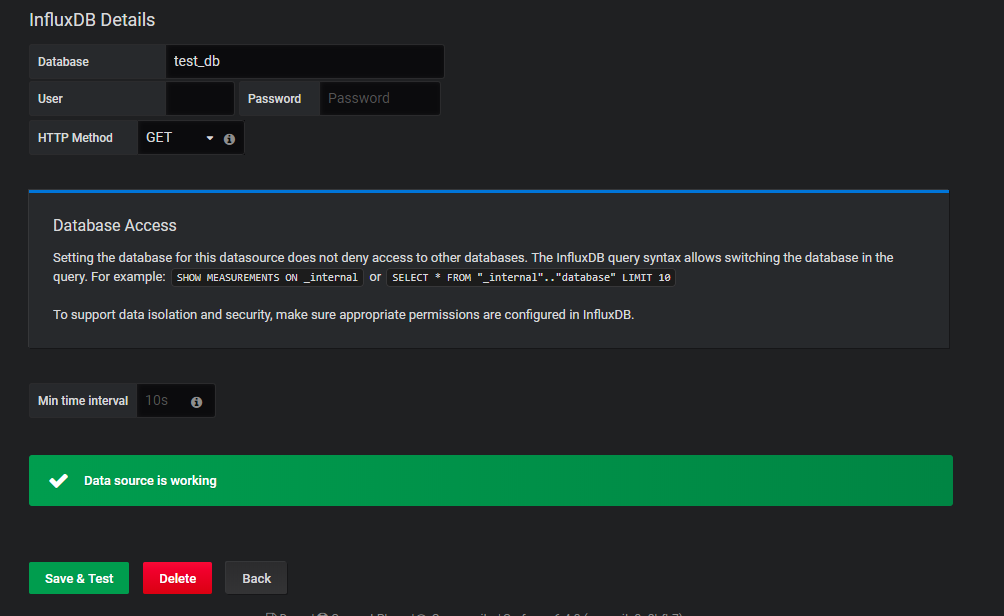
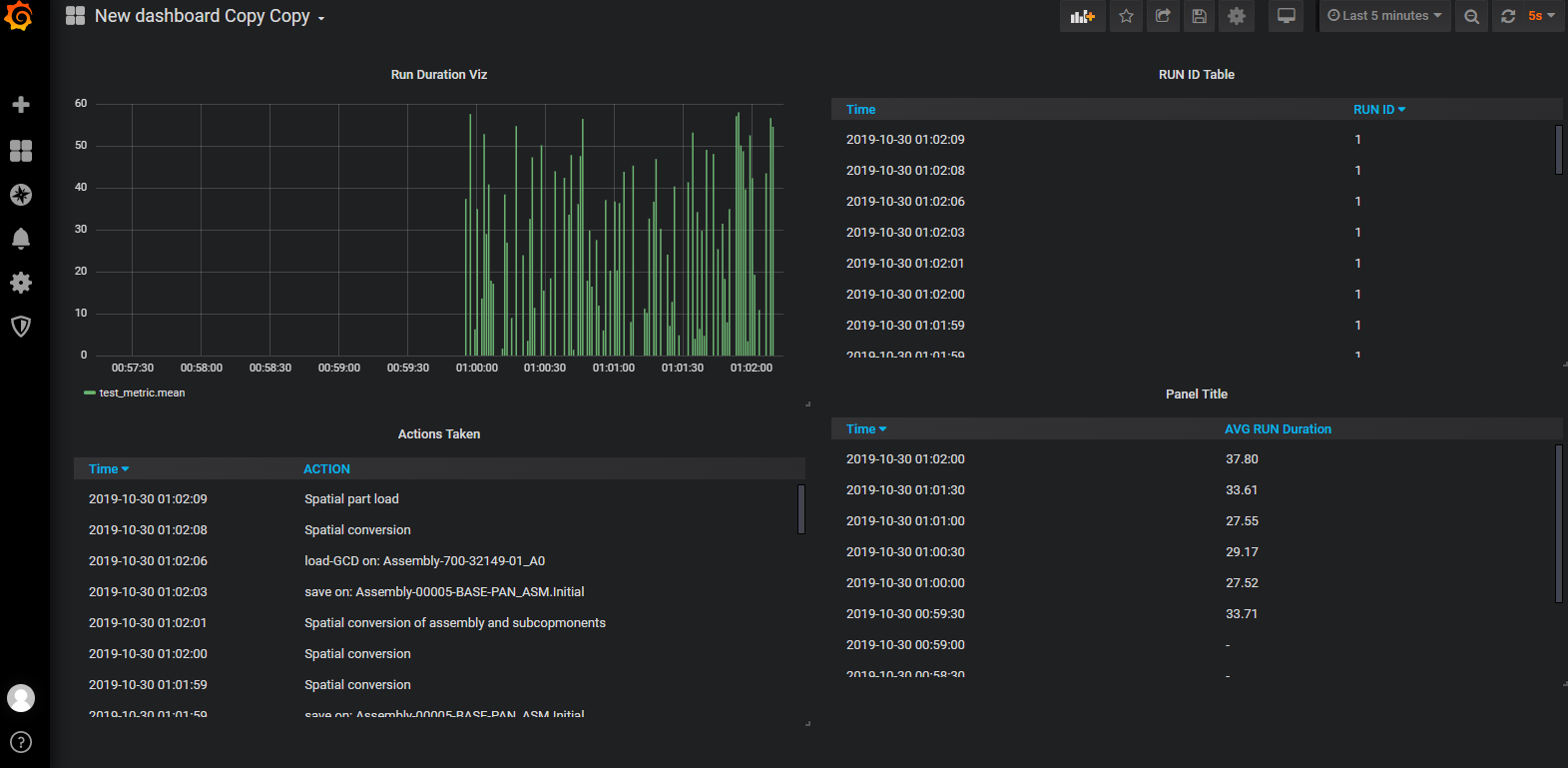
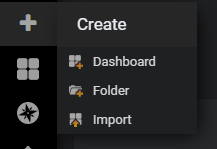
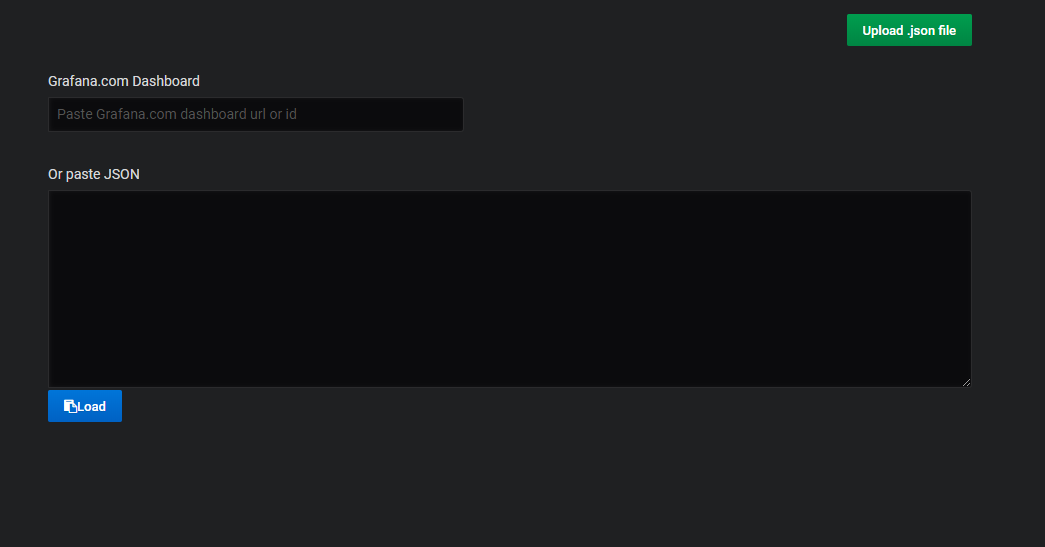
  database = "test\_db" # required

  ## Write timeout (for the InfluxDB client), formatted as a string.

  timeout = "5s"

* The CUSTOM\_LOG is tracking the different parameters namely Timestamps, Status, Action, Duration and Run ID
* Before we start the telegraf process we need to initiate the Influxdb so that the endpoint

["http://localhost:8086"] is available to store the incoming data

* Open the Influxdb folder which contains the following: 
* Open command prompt in this folder and enter the influx.exe and press enter, this will show the following: 
* This confirms Influxdb process has been initiated (don’t close this window)
* Now we need to generate the log files (mock log for testing) to simulate the streaming behavior
* Copy the attached python files (log\_generator.py and inject\_riun\_id.py) to a folder
* Inside log\_generator.py the no. of logs to be generated is defined by N, we can set the number higher to get keep generating logs at random intervals (1000 to 5000)
* Open command prompt on the folder with python files, and enter python log\_generator.py to start
* Next initiate the telegraf process by entering the following   
  telegraf.exe --config telegraf\_test.conf –test
* We will receive something similar to the following:  
  2019-10-29T19:11:29Z I! Starting Telegraf 1.12.3
* If we didn’t receive the above mentioned prompt probably some of the steps has been missed and we need to troubleshoot before we can continue.
* If did receive that prompt then we can start the actual telegarf process by the following:  
  telegraf.exe --config telegraf\_test.conf
* To check if we are receiving data on influxdb open the influxdb folder again
* Open a command prompt and type influx.exe and press enter, this will show following:  
  
* Next enter SHOW DATABASES  
  
* Our test\_db should show up here
* Next USE test\_db  
  
* Next SHOW MEASUREMENTS it should show test\_metric  
  
* Next SHOW FIELD KEYS  
  
* Next SELECT \* FROM test\_metric LIMIT 5  
  
* This ensures we are receiving data on influxdb from telegraf
* Now we need to start Grafana, open the Grafana folder and go to bin folder
* Double click on grafana-server.exe or open through command prompt
* The Web GUI is at <http://localhost:3000> open browser and go to this url
* The Grafana login page will appear, the default user/pass is admin/admin (change if you want to on next page)
* Now we need to setup the influxdb input for grafana
* Click on the gear icon on the sidebar and select Data   
  
* On next window select Add data source  
  
* Choose InfluxDb  
  
* On next window, add <http://localhost:8086> to the URL field our influxdb default endpoint and test\_db to the Database field  
    
  
* Once you click Save & Test it should be green indicating everything is working fine
* Next to setup the same dashboard as the following dashboard  
  
* Click on the plus icon and then Import  
  
* This will show the following window  
  
* Open the attached Grafana\_dashboard.json file copy everything inside and paste into the paste JSON box and click Load
* Next window click Import.
* That should give you the same dashboard setup.