**URL for EMQX enterprise:**

<http://localhost:18083/>

**SQL code while ceating rule in emqx to connect Kafka**

SELECT

payload.sensor AS sensor\_name,

payload.temperature AS temperature\_value,

payload.humidity AS humidity\_value,

payload.timestamp AS event\_time

FROM

"test-in"

**Commands to create Kafka topic and run kafka consumer**

bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

bin\windows\kafka-server-start.bat .\config\server.properties  
 To create topic

bin\windows\kafka-topics.bat --create --topic test-in –replication-factor 1 –partitions 3 --bootstrap-server localhost:9092

To retrieve the existed topic

bin\windows\kafka-topics.bat --describe --topic test-in --bootstrap-server localhost:9092

bin\windows\kafka-console-consumer.bat --topic test-in --bootstrap-server localhost:9092

**Commands to run Influxdb and telegraf**

C:/ProgramFiles/../InfluxData/Influxdb/influxd.exe

C:/ProgramFiles/../InfluxData/Telegraf/telegraf.exe –config telegraf.conf

**To run influxdb the URL**

<http://localhost:8086/>

**General Query:**

from(bucket: "Project")

  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)

  |> filter(fn: (r) => r["\_measurement"] == "kafka\_consumer")

  |> filter(fn: (r) => r["\_field"] == "humidity\_value" or r["\_field"] == "temperature\_value")

  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)

  |> yield(name: "mean")

**Query 1 which is for average for 5 minutes**

from(bucket: "Project")

  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)

  |> filter(fn: (r) => r["\_measurement"] == "kafka\_consumer")

  |> filter(fn: (r) => r["\_field"] == "humidity\_value" or r["\_field"] == "temperature\_value")

  |> aggregateWindow(every: 5m, fn: mean, createEmpty: false)

  |> yield(name: "mean")

**Query 3a For Daily average or offline**

from(bucket: "Project")

|> range(start: -30d)

|> filter(fn: (r) => r["\_measurement"] == "kafka\_consumer")

|> filter(fn: (r) => r["\_field"] == "temperature\_value" or r["\_field"] == "humidity\_value")

|> filter(fn: (r) => r["host"] == "Teja\_M")

|> aggregateWindow(every: 1d, fn: mean, createEmpty: false)

|> yield(name: "daily\_mean")

**Query 3b For Rolling Average**

from(bucket: "Project")

|> range(start: -7d)

|> filter(fn: (r) => r["\_measurement"] == "kafka\_consumer")

|> filter(fn: (r) => r["\_field"] == "temperature\_value")

|> filter(fn: (r) => r["host"] == "Teja\_M")

|> movingAverage(n: 12)

|> yield(name: "rolling\_average")