Monitoring Solution - Influx, CollectD, Grafana & JMeter

PART 1 : Complete Monitoring Solution for JMeter and Load Generators:
PART 2 : Enable this configuration for JVM monitoring

1. **Install Influx DB**
    a. On Mac:
        i. brew update
        ii. brew install influxdb
        iii. ln -sfv /usr/local/opt/influxdb/*.plist ~/Library/LaunchAgents # Having launchctl to start influxDB at login
        iv. launchctl load ~/Library/LaunchAgents/homebrew.mxcl.influxdb.plist # Using launchctl to start influxDB
        v. brew services start influxdb
    b. On Ubuntu:
        i. sudo apt-get update
        ii. sudo curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add
        iii. sudo echo "deb https://repos.influxdata.com/ubuntu trusty stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
        iv. sudo apt-get update
        v. cat /etc/apt/sources.list.d/influxdb.list
        vi. sudo cat /etc/apt/sources.list.d/influxdb.list
        vii. sudo apt-get upgrade
        viii. sudo vi /etc/apt/sources.list.d/influxdb.list
        ix. sudo apt-get update
        x. sudo apt-get -y isntall influxdb
        xi. ps -ef | grep influx
        xii. netstat -aeno | grep 136935
2. Once your are done with the installation using the command - 1.b.x, using the below two commands we can understand on which port influx DB is running by using 1.b.xi and 1.b.xii
3. After getting the port on which this application is running we now need to connect to the influxDB and create a database to store all the data (As a part 1 activity we are creating 2 databases - 1 for storing JMeter reports and 2 for storing VM's monitoring stats)
        i. influx --------- Using this command we are connected to the Influx Database
        ii. show databases --- This command will show all the database name which are present in INFLUX DB - By Default "_internal" database is present.
        iii. create database <database_name>
        Ex:
        create database stats_client_jmeter ---- "stats_client_jmeter" is database to store jmeter related statistics
        create databases stats_server_collectD ---- "stats_server_collectD" is database to store system monitoring information like CPU, MEMORY etc..

iv. use <database> ----- This command is used to connect to the particular dB
Remember : Influx DB is a time series database, when ever we insert any data in the Influx DB it is indexed with time series

v. sudo service influxdb restart -------- Restart of the services

4. **Installing "COLLECTD"** package to capture system level stats
   a. sudo apt-get update
   b. sudo apt-get install collectd
   c. Once collectD is installed using above statement we need to update the conf file present at "/etc/collectd/collect.conf".
   NOTE : This file is edited to get the host name on which monitoring is done and data capture policy along with the plugins or type of data which we want to use. Updating this local system configuration to capture CPU and Memory along with few more stats like DISK usage
   d. vi /etc/collectd/collectd.conf
   ---------- OPEN collectD conf file in any of the editor, note in collectD conf file we can specify the pulgins which are needed to specify the type of metrics for collection. By default CPU, Memory and other information are loaded we can add or remove the default one. Default setting at "collectd.conf"
   Remember : This is the most important part of configuration - when we are throwing ball someone should be there to catch it, keeping this part in mind lets say if collect d is going to capture the stats it has to push that data into influxdb and in this scenario INFLUX should be ready to catch it.
   Now, communication between the collectd and influxDB will be done over UDP protocol so our system should be able to establish communication over the same.
   sudo tcpdump -i eth0 -p -n dst port 8096 **→ If its listening with UDP on this port then we are good or else we can enable port to accept UDP communication**
   sudo iptables -A INPUT -p UDP –dport 8096 -j ACCEPT
   sudo iptables -A OUTPUT -p UDP –deport 8096 -j ACCEPT

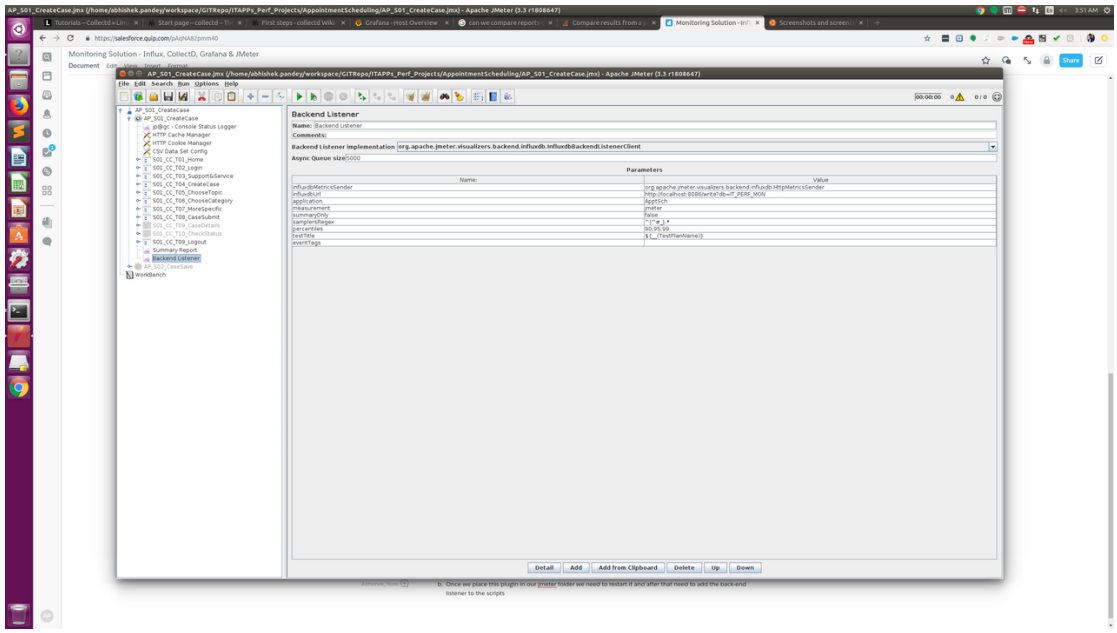
   Once this part is done we need to update the collectd.conf file to send data to INFLUXDB
   Step 1: Enable plugin:
   LoadPlugin network
   Step 2:
   <Plugin "network">
   Server "<InfluxDBServer>" "<port_no - 8096>"
   </Plugin>
   e. sudo service collectd status --- Check the status of collectD
   f. sudo service collectd stop ---- Starting the collectd service on the box

5. **Configuring InfluxDB to store data from collectD**
   a. Edit the influxdb.conf file --- CollectD part as provided below

```
[[collectd]]
  enabled = false
  bind-address = ":25826"
  database = "stat_server_collectd"
  retention-policy = ""
  # The collectd service supports either scanning a directory for multiple types
  # db files, or specifying a single db file.
  typesdb = "/usr/local/share/collectd/types.db"
  # Flush if this many points get buffered
  batch-size = 10000
  # Number of batches that may be pending in memory
```

batch-pending = 10
Flush at least this often even if we haven't hit buffer limit
batch-timeout = "10s"
UDP Read buffer size, 0 means OS default. UDP listener will fail if set above OS max.
read-buffer = 0

6. **Configuring Grafana**
   a. Update the “source.list” file and add the below line
      sudo vi /etc/apt/source.list --- Open the source.list file
      deb https://packagecloud.io/grafana/stable/debian/ stretch main ---- Add this particular line
   b. curl https://packagecloud.io/gpg.key | sudo apt-key add - ---- Get the signed packages
   c. sudo apt-get update
   d. sudo apt-get install grafana
   e. sudo service grafana-server start
      NOTE: By default grafana will be up and running with URL - http://:3000/
      And default password for login - admin/admin - for local linux VM - http://10.204.140.87/?orgId=1 user-admin
7. **Configuring JMeter to write data in InfluxDB:**
   a. For JMeter to write data to influxDB we need plugin top be placed in JMeter lib/ext folder
      JMeter-InfluxDB-Writer-plugin-1.2.jar
   b. Once we place this plugin in our jmeter folder we need to restart it and after that need to add the back-end listener to the scripts
      Once we select the back-end listener we need to provide the
      influx db url
      application name
      measurement as “jmeter”
      test value as ${__(TestPlanName)}
      All the information as shown in the image below



8. **<u>Configuring Grafana for displaying reports:</u>**

a. For displaying the report on grafana dashboard we first need to create data source
   datasource - it is the connection pool which establish the connection between influxDb and Grafana. In our local setup we have 2 data source
   i. collectd ---------- which is connecting to the “collectd” data base in influxDb
   ii. influxdb ---------- this datasource connects to “jmeter” database in influxdb
b. Once the datasource are created we need to import 2 pre-defined dashboards in grafana
   i. Hostover View - this dashboard will provide the system related matrices as shown in below image by selecting specific hostname
   ii. JMeter Dashboard (3.2 and UP) – this dashboard will provide all the data that we are capturing during run time of client side matrices from JMeter