Integrate Automation Performance Testing with Grafana

--- InfluxDB ---

1. Install InfluxDB using Docker container

`docker-compose.yml`

```
version: '3'
services:

  influxdb:
    image: influxdb:1.8
    container_name: influxdb
    restart: always
    ports:
      - 8086:8086
    volumes:
      - /etc/influxdb/influxdb:/var/lib/influxdb
      - /etc/influxdb/influxdb.conf:/etc/influxdb/influxdb.conf
```

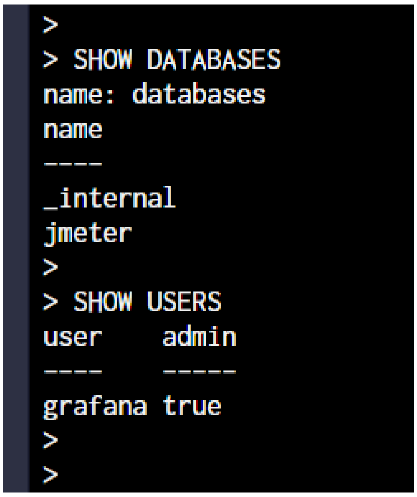
`/etc/influxdb/influxdb.conf`

```
[meta]
  dir = "/var/lib/influxdb/meta"

[data]
  dir = "/var/lib/influxdb/data"
  engine = "tsm1"
  wal-dir = "/var/lib/influxdb/wal"
[http]
  # Determines whether HTTP endpoint is enabled.
  enabled = true

  # The bind address used by the HTTP service.
  bind-address = ":8086"
```

2. Create Database , Create User, Grant User to use database (1 project per 1 DB)

`$ docker exec -it influxdb /bin/bash`

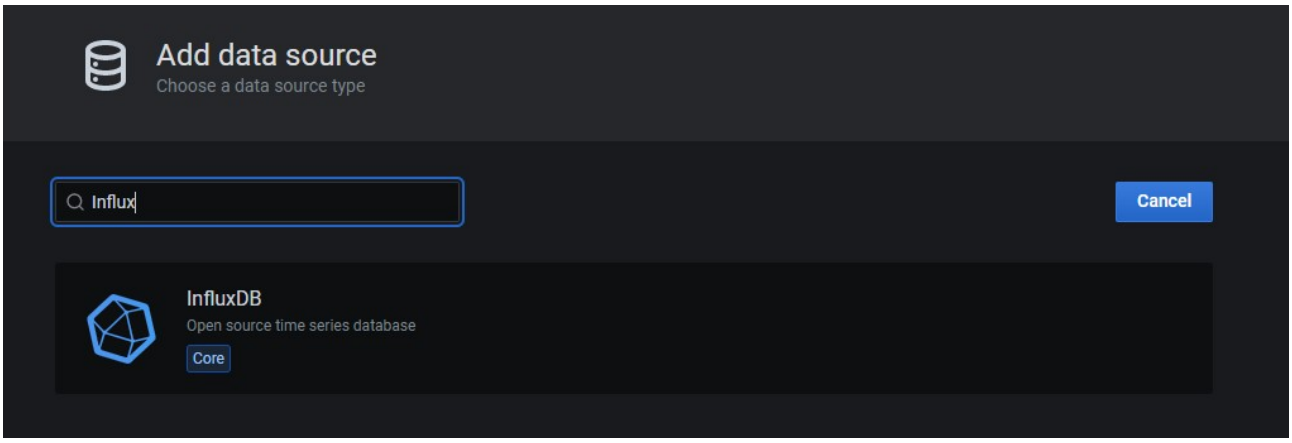
```
root@eec2c79c7757:/# influx

> CREATE USER grafana WITH PASSWORD '<password>' WITH ALL PRIVILEGES
> CREATE DATABASE jmeter1
> GRANT ALL ON jmeter1 to grafana

> SHOW DATABASES
> SHOW USERS
```



```
>
> SHOW DATABASES
name: databases
name
----
_internal
jmeter
>
> SHOW USERS
user    admin
----    -----
grafana true
>
>
```

--- Grafana ---

1. Add datasource from influxdb



Add data source

Choose a data source type

Influx

InfluxDB

Open source time series database

Query Language: InfluxQL

URL: http://<yourserver>:8086

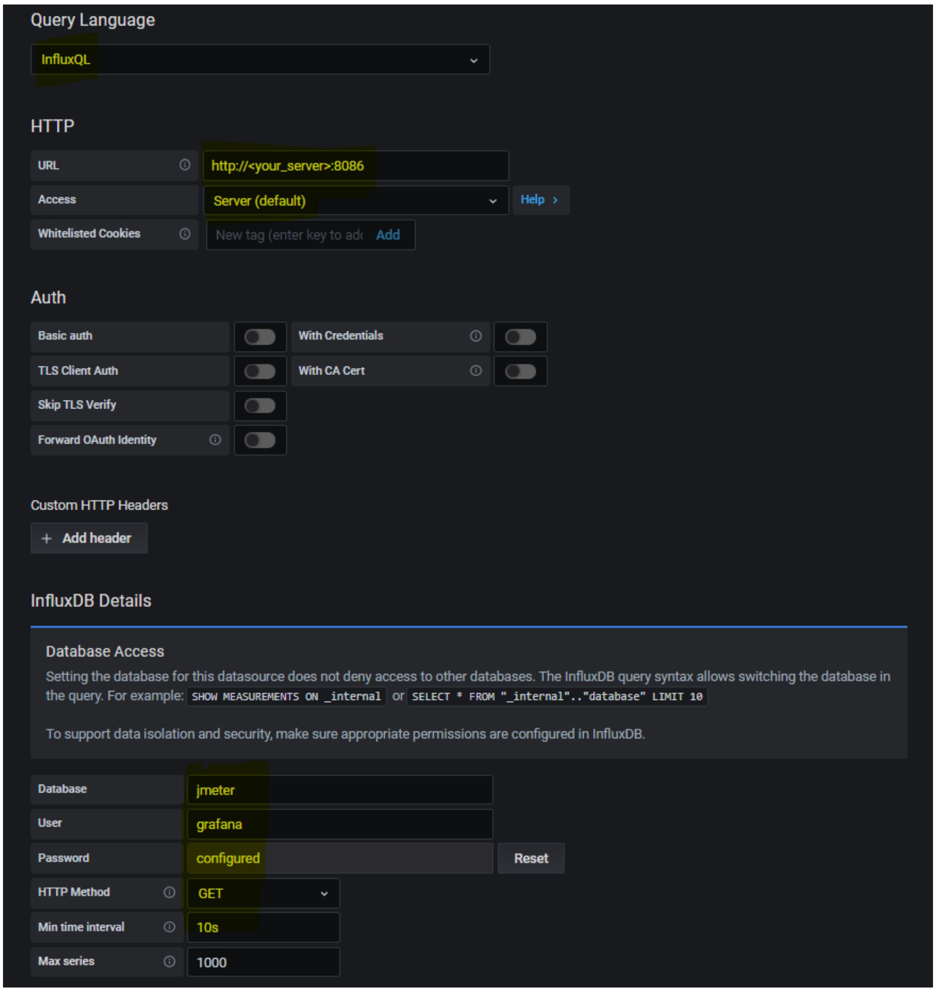
Turnoff all Auth

Database: jmeter

User: <DB_username>

Pass: <DB_password>

HTTP Method: GET



Query Language

InfluxQL

HTTP

URL: http://<your_server>:8086

Access: Server (default)

Whitelisted Cookies: New tag (enter key to ad

Auth

Basic auth

With Credentials

TLS Client Auth

With CA Cert

Skip TLS Verify

Forward OAuth Identity

Custom HTTP Headers

InfluxDB Details

Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal".."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database: jmeter

User: grafana

Password: configured

HTTP Method: GET

Min time interval: 10s

Max series: 1000

Click [Save & Test] to check connection

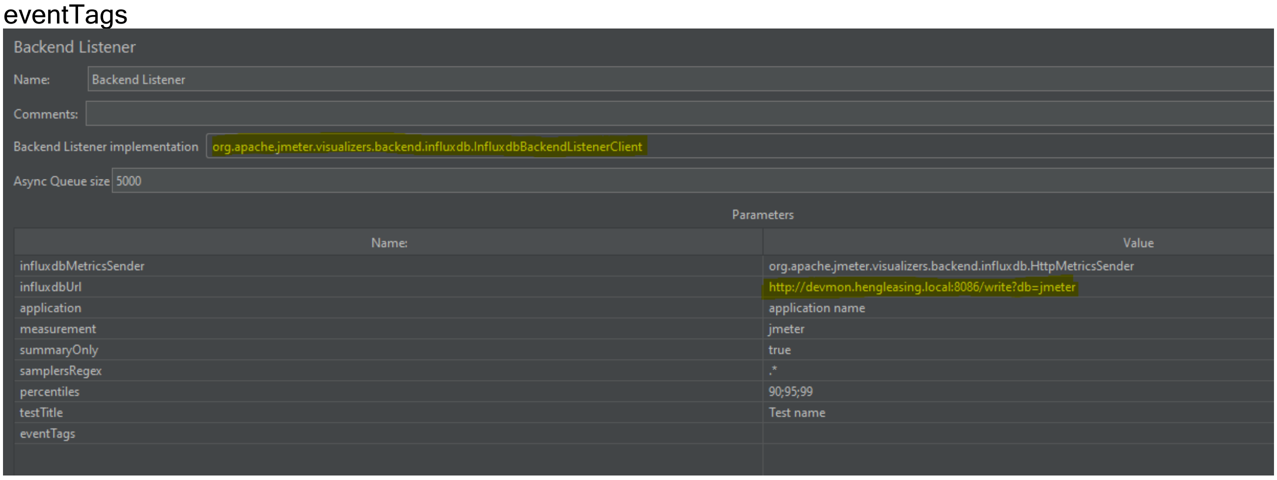
2. Add new Dashboard
Download JSON dashboard template from: https://grafana.com/grafana/dashboards/5496

--- jmeter ---

create .jmx script
1. Create your Test plan and Thread group
2. Create Backend Listener
Test Plan -> Add -> Listener -> Backend Listener

backend listener impletation: org.apache.jmeter.visualizers.backend.influxdb.InfluxdbBackendListenerClient

Parameters
influxdbMetricsSender: org.apache.jmeter.visualizers.backend.influxdb.HttpMetricsSender
influxdbUrl: http://<yourserver>:8086/write?db=<InfluxDB_Name>
application: application name
measurement: jmeter
summaryOnly: true
samplersRegex: .*
percentiles: 90;95;99
testTitle: Test name
eventTags



Backend Listener

Name: Backend Listener

Comments:

Backend Listener implementation org.apache.jmeter.visualizers.backend.influxdb.InfluxdbBackendListenerClient

Async Queue size 5000

Parameters

Name:	Value
influxdbMetricsSender	org.apache.jmeter.visualizers.backend.influxdb.HttpMetricsSender
influxdbUrl	http://devmon.hengleasing.local:8086/write?db=jmeter
application	application name
measurement	jmeter
summaryOnly	true
samplersRegex	.*
percentiles	90;95;99
testTitle	Test name
eventTags	

3. save jmx file and run performance test from anywhere