



Hőmérsékleti szenzorok adatainak vizualizációja és monitorozása

Kiberfizikai rendszerek

Mérési útmutató

Készítette: Vörös András, Szabó Richárd, Burján Dezső

Verzió: 1.1

2019.

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1. Általános tudnivalók

1.1. Grafana

- <http://docs.grafana.org/guides/basic-concepts/>
- <http://docs.grafana.org/features/panels/graph/>
- <http://docs.grafana.org/alerting/notifications/>
- <http://docs.grafana.org/alerting/notifications/#webhook>
- <http://docs.grafana.org/alerting/rules/>
- <http://docs.grafana.org/reference/timerange/>
- <http://docs.grafana.org/features/datasources/influxdb/>

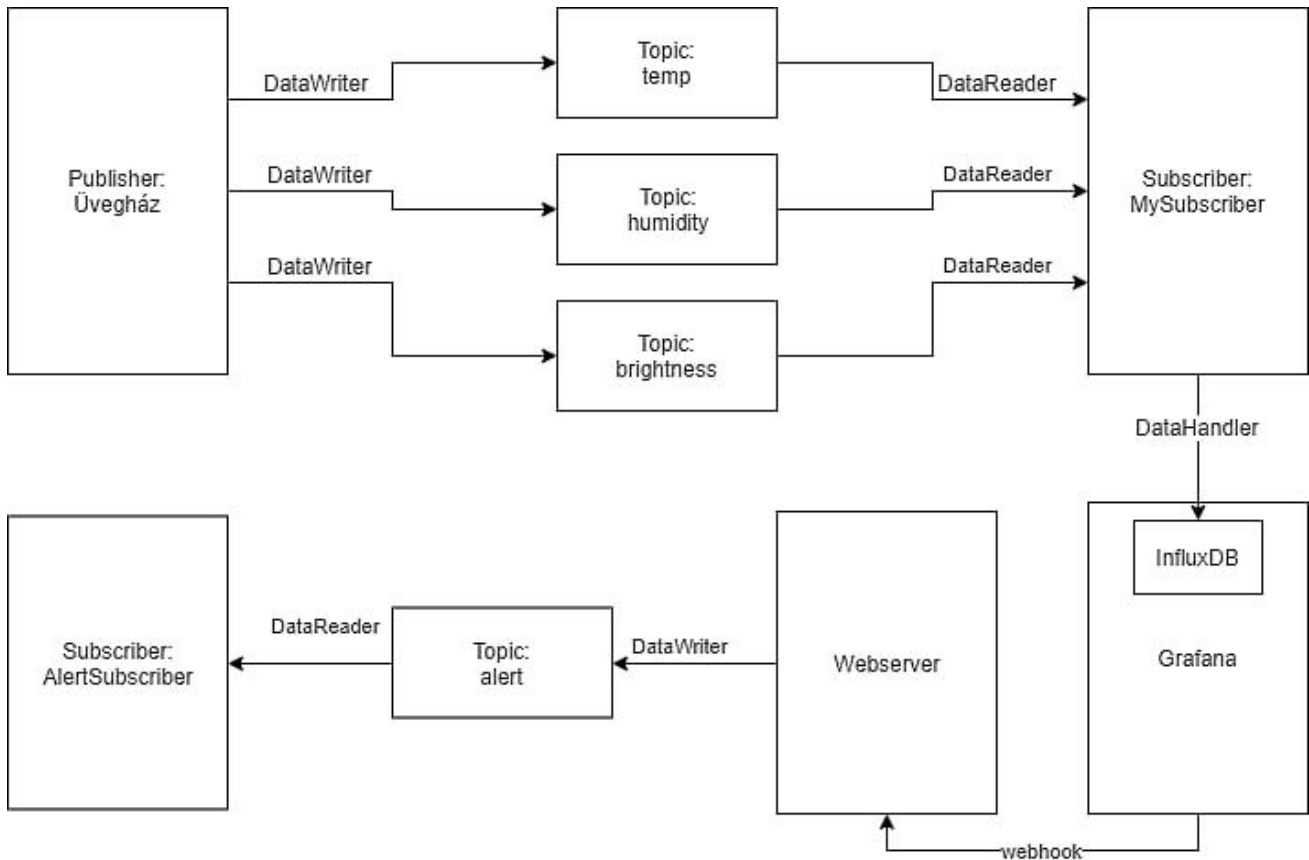
1.2. InfluxDB

- <https://docs.influxdata.com/influxdb/v1.7/concepts/key-concepts/>
- <https://docs.influxdata.com/influxdb/v1.7/query-language/data-exploration/>
- <https://docs.influxdata.com/influxdb/v1.7/query-language/data-exploration/#select-data-that-have-a-specific-tag-key-value>
- <https://docs.influxdata.com/influxdb/v1.7/query-language/database-management/>
- <https://docs.influxdata.com/influxdb/v1.7/query-language/functions#moving-average>

2. Szoftverkörnyezet

- **Eclipse**
- **RTI Connex**
- **Grafana:** Egy adatvizualizációra és monitorozásra használt eszköz. Különböző adatforrásokat lehet hozzáadni és dashboardokat lehet benne létrehozni a különböző grafikonoknak. Illetve riasztások és értesítési csatornák segítségével jelezni tudja, ha egy megjelenítőn valami nem kívánt anomália lépett fel vagy szűnt meg.
- **InfluxDB:** Egy idősorok tárolására használt adatbáziskezelő rendszer, amely sokban hasonlít a közkezdvelt MySQL-re.

3. Szenzor adatok elérése



A szenzorból adatokat a DDS segítségével nyerhetünk. Jelen labor alatt egy előre felvett mérés visszajátszását fogjuk végrehajtani, amellyel azt szimuláljuk, mintha az adatokat valós időben kapnánk a szerkezettől.

A példaként kiadott **Eclipse** projektben a teljes folyamat el van készítve, a DDS kommunikáció, az adatok InfluxDB-be történő kiírása.

Az InfluxDB a cps adatbázisba dolgozik, azon belül található egy cps nevű measurement (analógia a relációs adatbázisban **tábla**). Van egy **tag** (oszlopok, amelyek mindig karakterláncok, ezekre épít indexet az adatbáziskezelő rendszer, így ezekre érdemes keresni) értéke, aminek a neve **name**. És egy **field** (erre csak lineáris kereséssel tudunk keresni) valós szám, ez reprezentálja a hőmérsékletet, a neve pedig value.

Példa adatbázis

```
cps,name=temp0 value=27.9 <TIMESTAMP>
cps,name=temp0 value=27.9 <TIMESTAMP>
cps,name=temp0 value=27.9 <TIMESTAMP>
cps,name=temp1 value=27.9 <TIMESTAMP>
cps,name=temp2 value=27.9 <TIMESTAMP>
```

Példa lekérdezés: SELECT "value" FROM "cps" WHERE "name"='temp2'

4. Vizualizáció

A virtuális gép elindítása után a grafana és az influxdb service automatikusan elindul, a <http://localhost:3000> webcímen böngészőből érdemes használni. Az alapértelmezett bejelentkezési adatok: **admin/LaborImage**.

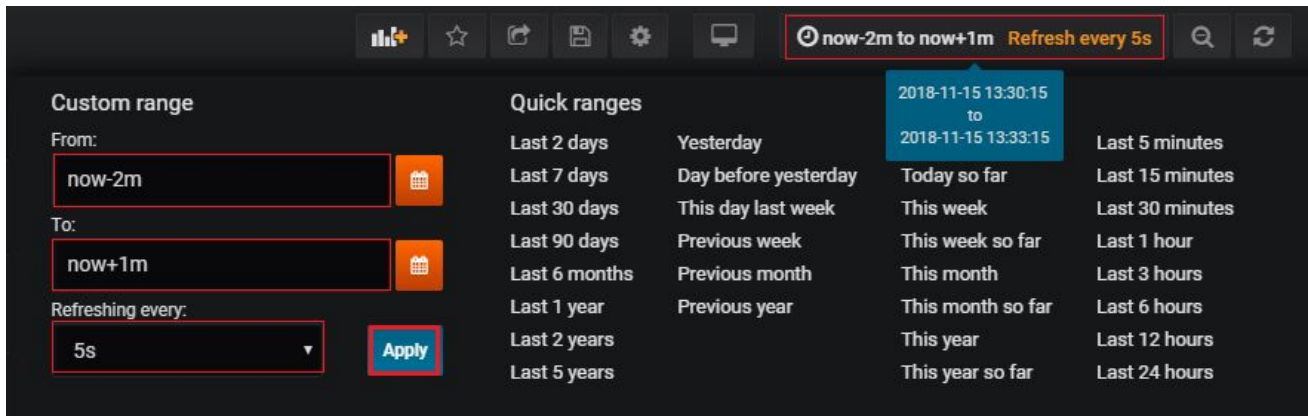
Indítsa el a Grafana szerveret és lépjen be a böngészőben a kliens oldalra. Az adatforrást előzetesen bekonfiguráltunk, hogy az InfluxDB legyen. Egy példa Dashboard is elérhető, amibe már be van állítva, hogy az InfluxDB-ből olvassa be a szenzor adatait.

Feladat

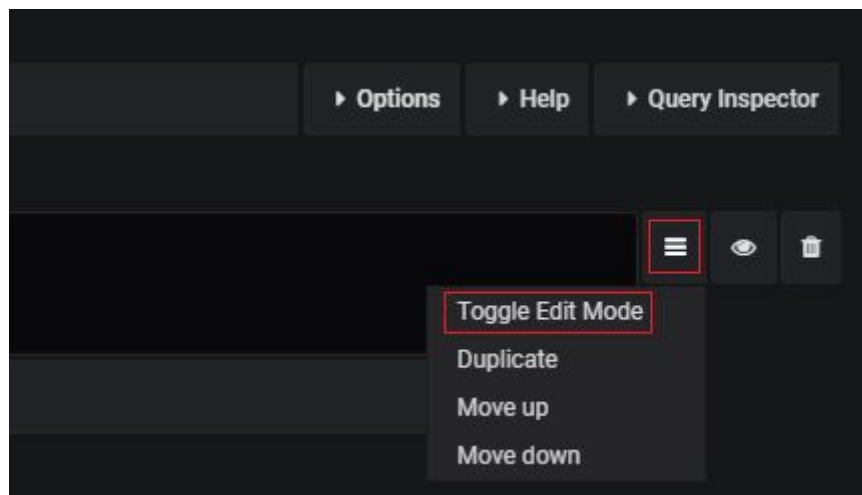
Nyissa meg az Eclipse projektet és módosítsa úgy, hogy miután a *Subscriber* megkapta az adatot, ne csak akkor küldje tovább az InfluxDB-nek, ha a küldő szenzor azonosítója **temp0**, hanem akkor is, ha **temp1** vagy **temp2**, ezzel három szenzor adatait fogjuk egyszerre megkapni. A használható azonosítókat kérdezze meg a labor vezetőjétől.

A Grafana felületen hozzon létre egy új Dashboardot vagy használja a példát, adjon hozzá 6 darab panelt (*Graph* panelek legyenek). Állítsa be, hogy az adatforrásuk az **InfluxDB** legyen.

Ha új Dashboardot hozott létre állítsa be, hogy a figyelt intervallum folyamatosan frissüljön.

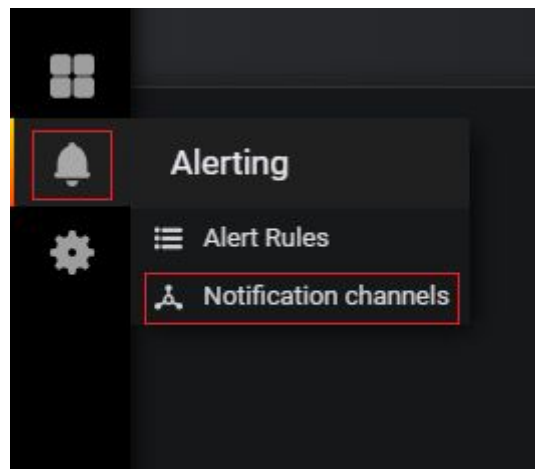


Külön-külön mind a háromra állítsa be, hogy az az egyes szenzorok adatait mutassák (tehát panel1 és panel2 akkor, ha pl. temp0 az azonosító, panel3 és panel4 akkor, ha pl. temp1 az azonosító). Érdemes az Influx lekérdezést leírni az összekattingatás helyett.



5. Monitorozás

A Grafana felületen meg lehet adni, hogy az egyes **Alert**eket hova küldje el a szerver, a labor környezetben ezt már előre bekonfiguráltunk egy **webhook**ot, melynek lényege, hogy egy POST vagy PUT kérést küld egy adott webcímre (jelen esetben ez a <http://localhost:6354/alert> és POST kérés, a webszerver, amely kiszolgálja ezt a kérést megtalálható a kiadott kódban a **ServerMain.java** fájl az, ennek indításával fogadhatjuk az **Alert**eket és DDS-en keresztül ki is tudjuk őket küldeni).



A Grafana egy hátránya, hogy csupán egy Alertet lehet létrehozni Panelenként és az ÉS/VAGY feltételeket sem szereti igazán, de nem komplex monitorozásra hozták létre, inkább a vizualizáció a fontosabb. Ha valaki komplexebb monitorozót keres, az használjon **Prometheust**.

Feladat 1

Hozzon létre kettő-kettő új riasztást minden panelhez, amelynél megadja, hogy küldjön riasztást, ha:

- A lekérdezés értékének utolsó eleme kisebb, mint 19 (**panel1, panel3, panel5**)
- A lekérdezés értékének utolsó eleme nagyobb, mint 25 (**panel2, panel4, panel6**)

Ezzel nagyjából azt tudjuk szimulálni, hogy riasztást adunk ki, ha az üvegházban az egyik szenzor elromlott vagy valami miatt nem megfelelő értéket ad vissza.

Állítsa be, hogy a Notification channel az előre bekonfigurált webhook legyen. A notification üzenete a neve legyen, lehetőleg ékezetek nélkül és valami azonosító pl. panelX_alert mondjuk szóközzel elválasztva a kettőt.

Graph General Metrics Axes Legend Display Alert Time range

Alert Config Alert Config

Notifications (1) Name Panel Title alert Evaluate every 1s

State history

Delete

Conditions

WHEN last () OF query (A, 5m, now) IS BELOW 19

+

If no data or all values are null SET STATE TO No Data ▾

If execution error or timeout SET STATE TO Alerting ▾

Test Rule

A lenti ábra vízszintes piros sávjában láthatjuk a riasztás kritikus sávját, ha bármelyik érték belép oda, akkor egy critical alert jön létre, ezt reprezentálja a szaggatott piros függőleges vonal, a zöld pedig, hogy az adat kilépett a kritikus állapotból és újra normál.



Feladat 2

Adjon hozzá két új Graph panelt a Dashboardhoz és kérje le mindenféle szűrés nélkül a value előző 5 értékének mozgását. Állítsa be a következő riasztásokat:

- A lekérdezés értékének utolsó eleme kisebb, mint 19 (**panel7**)
- A lekérdezés értékének utolsó eleme nagyobb, mint 25 (**panel8**)

Ezzel nagyjából azt tudjuk szimulálni, hogy riasztást adunk ki, ha az üvegházban a hőmérséklet nem felel meg az általunk elvártaknak.