

Projekat 3 – Uputstvo

EdgeX

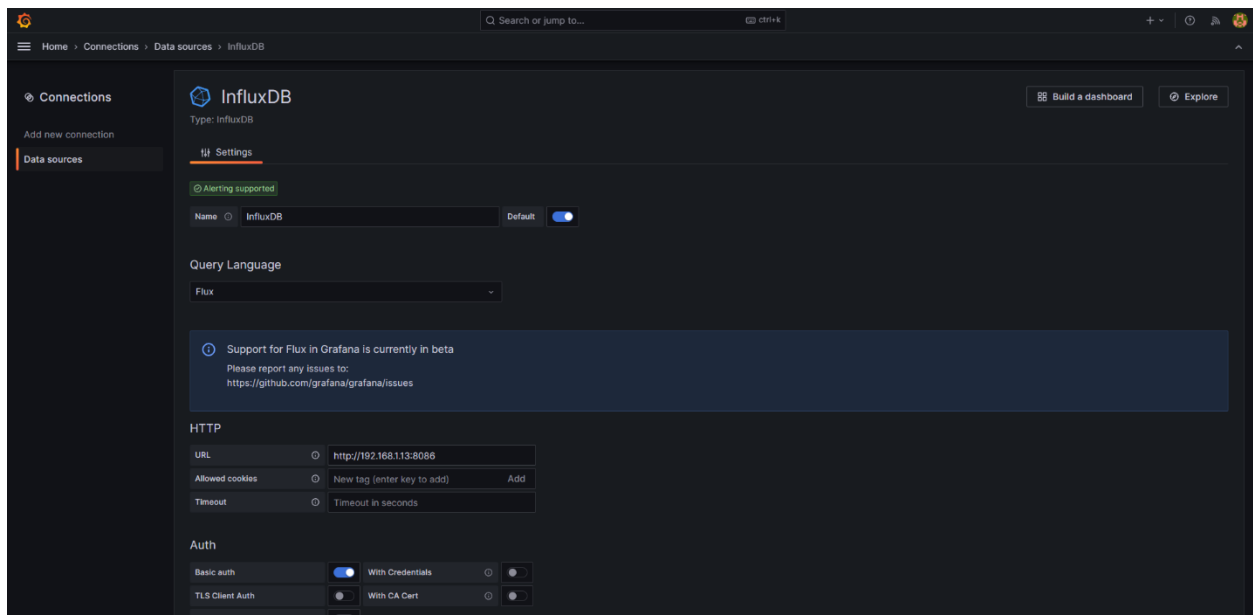
Za postavljanje EdgeX-a potrebno je da se konfiguriše izgled podataka koji dolaze, Device Profile, i Device. U dokumentu *Uputstvo.txt* su navedene sve HTTP metode koje treba da se pozovu sa odgovarajućim sadržajem. Što se tiče MQTT borkera, koristio sam javno dostupni **broker.hivemq.com** na portu **1883**. Topic na koji EdgeX šalje sve podatke koji mu stignu je **edgex/sensor_value**. Skripta *script.py* iz *data-rader-service* foldera se koristi za simulaciju uredjaja koji šalje podatke EdgeX-u.

Monitoring service

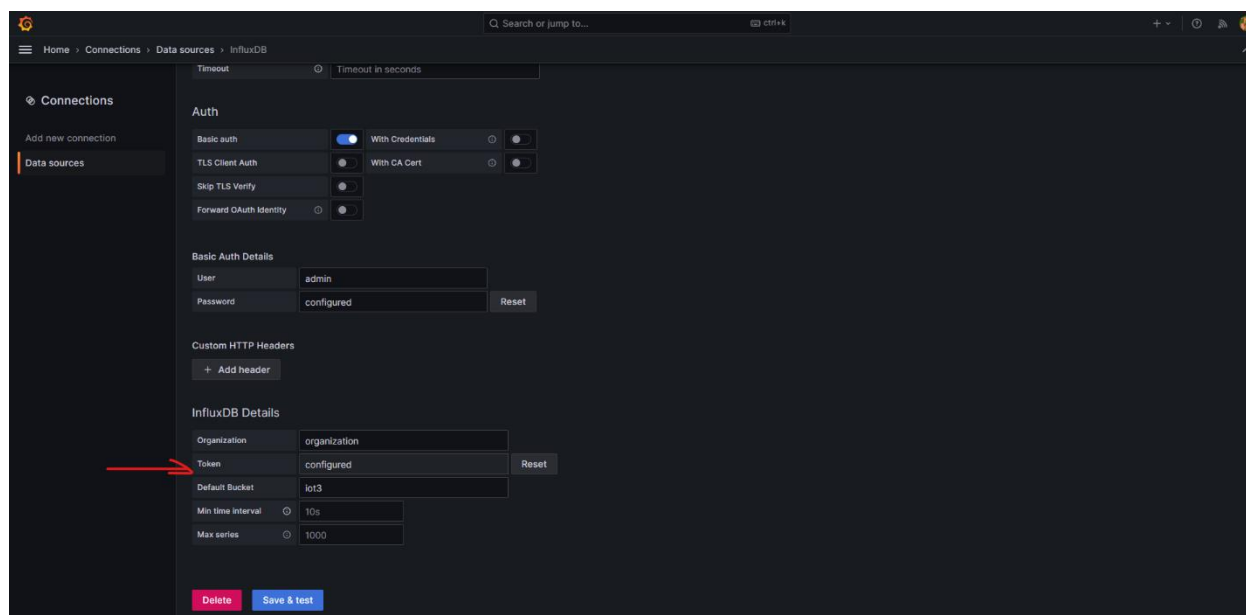
Monitoring service je pisan u .Net-u. Cilj mu je da prikupi podatke sa MQTT brokera, i da ih smesti u InfluxDB, gde će se korišćenjem Grafane vizualizovati podaci.

InfluxDb i Grafana

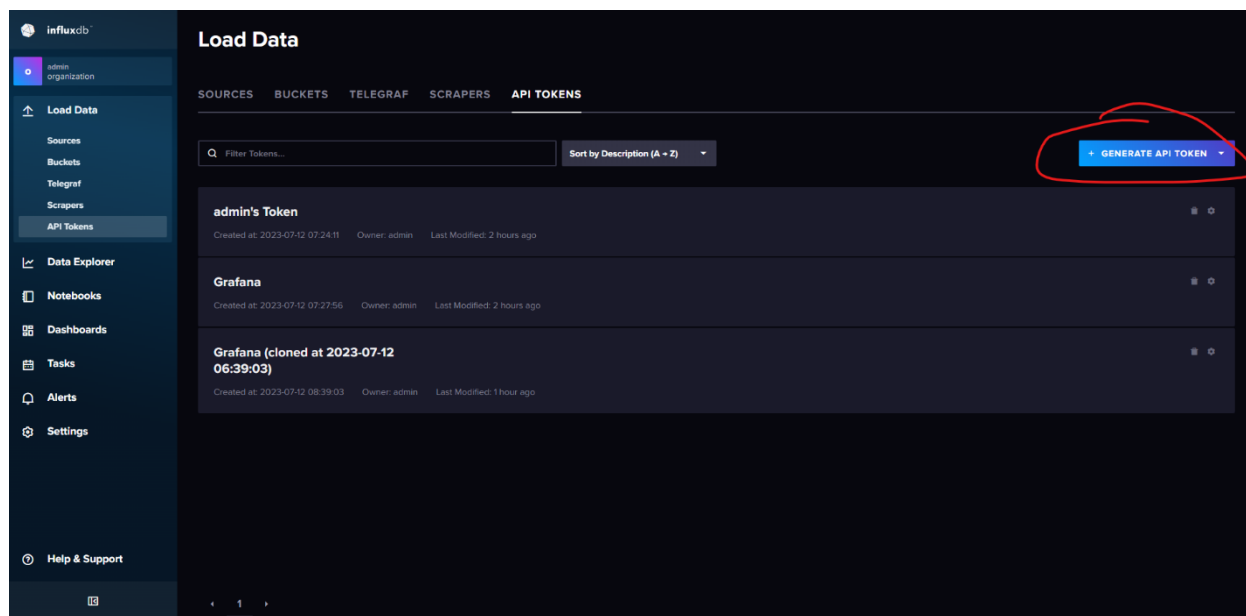
Pri kreiranju naloga na InfluxDB-u na portu 8086, potrebno je uneti username **admin**, password **adminadmin**, organizaciju **organization** i bukcet **iot3**. Nakon toga potrebno je postaviti grafanu. Inicijalni username i password za grafanu je admin, ali kasnije zahteva da se izmeni. Kod Grafane, potrebno je napraviti konekciju sa InfluxDB kao sa slike. Kod URL-a je potrebno staviti



IP adresu mašine na kojoj su pokrenuti docker kontejneri. Da bi se dobio token koji je potreban



kod InfluxDB Details dela, potrebno je otići u InfluxDB i generisati novi All Access Token.



Zatim je potrebno napraviti **Dashboard** koji će vizualizovati podatke iz konekcije koju selektujemo.

Nakon toga, potrebno je otići u InfluxDB i selektovati podatke koji želimo da vizualizujemo, i kopirati skriptu u Grafani.

Home > Dashboards

Dashboards

Playlists

Snapshots

Library panels

Dashboards

Create and manage dashboards to visualize your data

Search for dashboards and folders

Filter by tag

Starred

Sort

New

New Dashboard

New Folder

Import

General

InfluxDB

admin organization

Load Data

Data Explorer

Notebooks

Dashboards

Tasks

Alerts

Settings

Help & Support

Data Explorer

Graph

CUSTOMIZE

Local

SAVE AS

Looks like you don't have any queries. Be a lot cooler if you did!

Query 1

View Raw Data

CSV

Past 1h

SCRIPT EDITOR

SUBMIT

FROM

Search buckets

iot3

monitoring

tasks

Create Bucket

Filter

_measurement

temperature

Filter

_field

celsius_degrees

No tag keys found in this current time range.

WINDOW PERIOD

CUSTOM

AUTO

auto (10s)

Fill missing values

AGGREGATE FUNCTION

CUSTOM

AUTO

mean

median

last

Query 1

View Raw Data

CSV

Past 1h

QUERY BUILDER

SUBMIT

```
1 from(bucket: "iot3")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "temperature")
4   |> filter(fn: (r) => r["_field"] == "celsius_degrees")
5   |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
6   |> yield(name: "mean")
```

Filter Functions...

Transformations

aggregate.rate

changeMomentumOscillator

columns

cov

covariance

cumulativeSum

date.hour

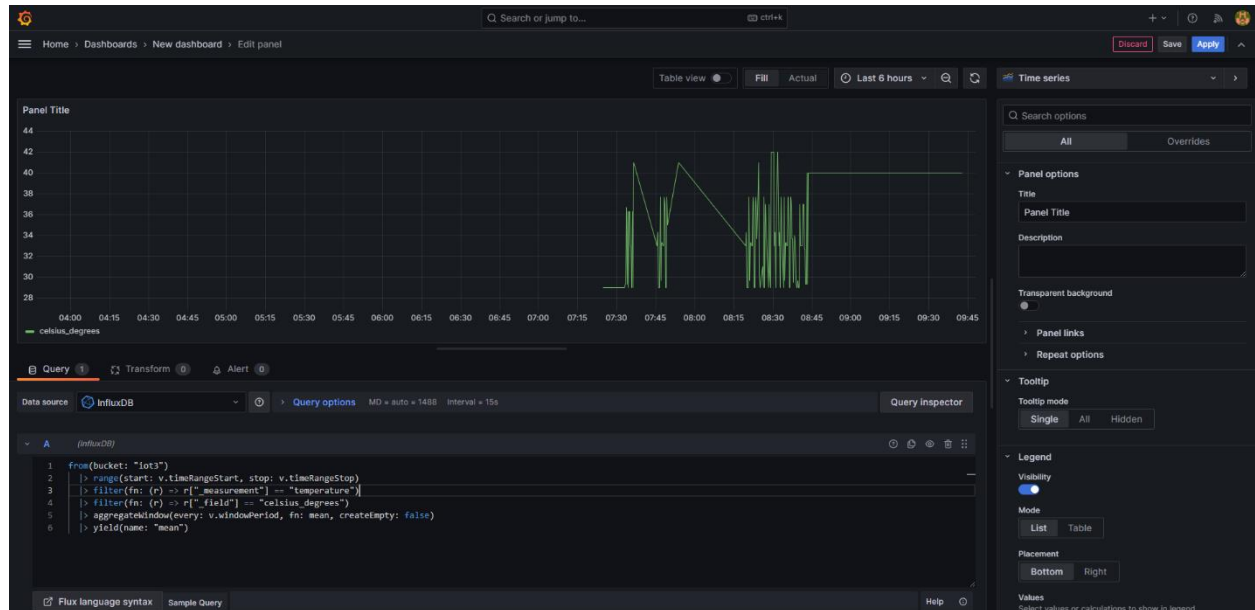
date.microsecond

date.millisecond

Functions

Variables

Nakon toga treba da se vidi ovako nešto na ekranu.



Visualization service

Visualization service je pisan u node.js-u. Ima za cilj da skupi podatke koje EdgeX šalje na MQTT topic, da detektuje situaciju ukoliko je temperatura van opsega, i da obavesti Device preko EdgeX-a o potrebnim akcijama koje treba da preduzme.

Python skripta

Python skriptu je potrebno pokrenuti nakon pokretanja celog sistema. Sistem se pokreće pozivom **docker compose up --build** iz **root** foldera projekta, a skripta se pokreće pozivom **py script.py** iz **data-reader-service** foldera.

Arhitektura projekta

