

Projekat 3 – IoT Smart-Building

Cilj ovog projekta bila je implementacija dva mikroservisa u različitim tehnologijama, kao i njihova integracija sa EdgeX platformom. Ovi mikroservisi predstavljaju Northbound infrastrukturu na fog i cloud strani ove platforme. Oba mikroservisa startovana su kao Docker kontejneri, što omogućava njihovu jednostavnu i skalabilnu implementaciju.

Prvi mikroservis je **Visualization mikroservis** koji ima zadatak da dobija podatke iz EdgeX platforme preko MQTT protokola. Za ovu svrhu, korišćen je MQTT broker HiveMQ. Nakon prijema podataka, Visualization mikroservis smešta podatke u InfluxDB bazu podataka. Da bi omogućio vizuelizaciju podataka smeštenih u bazi podataka, koristi se Grafana, popularan alat za vizualizaciju podataka. Grafana je pokrenuta kao Docker kontejner i pruža interfejs za pregled i analizu podataka iz InfluxDB baze.

Drugi mikroservis je **Monitoring mikroservis** koji takođe pretplaćen na isti topic na MQTT brokeru kao i Visualization mikroservis. Na ovaj način i on dobija podatke sa EdgeX platforme. Monitoring mikroservis primenjuje jednostavno pravilo za detekciju događaja na osnovu dobijenih podataka. Kada detektuje događaj, šalje odgovarajuću poruku ili komandu mikroservisu na EdgeX platformi.

U nastavku ovog dokumenta će biti detaljnije opisani pojedini mikroservisi, tehnologije koje će biti korišćene, kao i koraci za njihovu implementaciju i integraciju sa EdgeX platformom.

EdgeX

Da bi se konfigurisao EdgeX, mora se prilagoditi format dolaznih podataka, postaviti profil uređaja i konfigurisati sam uređaj (Device).

Na Data servisu, potrebno je izvršiti HTTP POST metodu kako bi se kreirao ValueDescriptor:
adresa: <http://localhost:48080/api/v1/valuedescriptor>

```
body1:
{
  "name": "id",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "String",
  "uomLabel": "id",
  "defaultValue": "null",
  "labels": [
    "environment",
    "id"
  ]
}
```

body2:

```
{
  "name": "timestamp",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "String",
  "uomLabel": "timestamp",
  "defaultValue": "null",
  "labels": [
    "environment",
    "timestamp"
  ]
}
```

body3:

```
{
  "name": "co2",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "String",
  "uomLabel": "co2",
  "defaultValue": "null",
  "labels": [
    "environment",
    "co2"
  ]
}
```

body4:

```
{
  "name": "temperature",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "Int64",
  "uomLabel": "temperature",
  "defaultValue": "null",
  "labels": [
    "environment",
    "temperature"
  ]
}
```

```
body5:
{
  "name": "light",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "String",
  "uomLabel": "light",
  "defaultValue": "null",
  "labels": [
    "environment",
    "light"
  ]
}
```

```
body6:
{
  "name": "humidity",
  "description": "data from .csv file",
  "min": "",
  "max": "",
  "type": "String",
  "uomLabel": "humidity",
  "defaultValue": "null",
  "labels": [
    "environment",
    "humidity"
  ]
}
```

Nakon ovoga, sledeći korak je kreiranje Device Profile-a pozivanjem HTTP POST metode na adresi <http://localhost:48081/api/v1/deviceprofile/uploadfile>, i selektovanjem fajla docker-profile.yaml

Zatim je potrebno kreirati device HTTP POST metodom na <http://localhost:48081/api/v1/device>:

```
{
  "name": "SensorValueCluster2",
  "description": "Raspberry Pi sensor cluster",
  "adminState": "unlocked",
  "operatingState": "enabled",
  "protocols": {
    "HTTP": {
      "host": "localhost",
      "unitID": "1"
    }
  },
  "labels": [
```

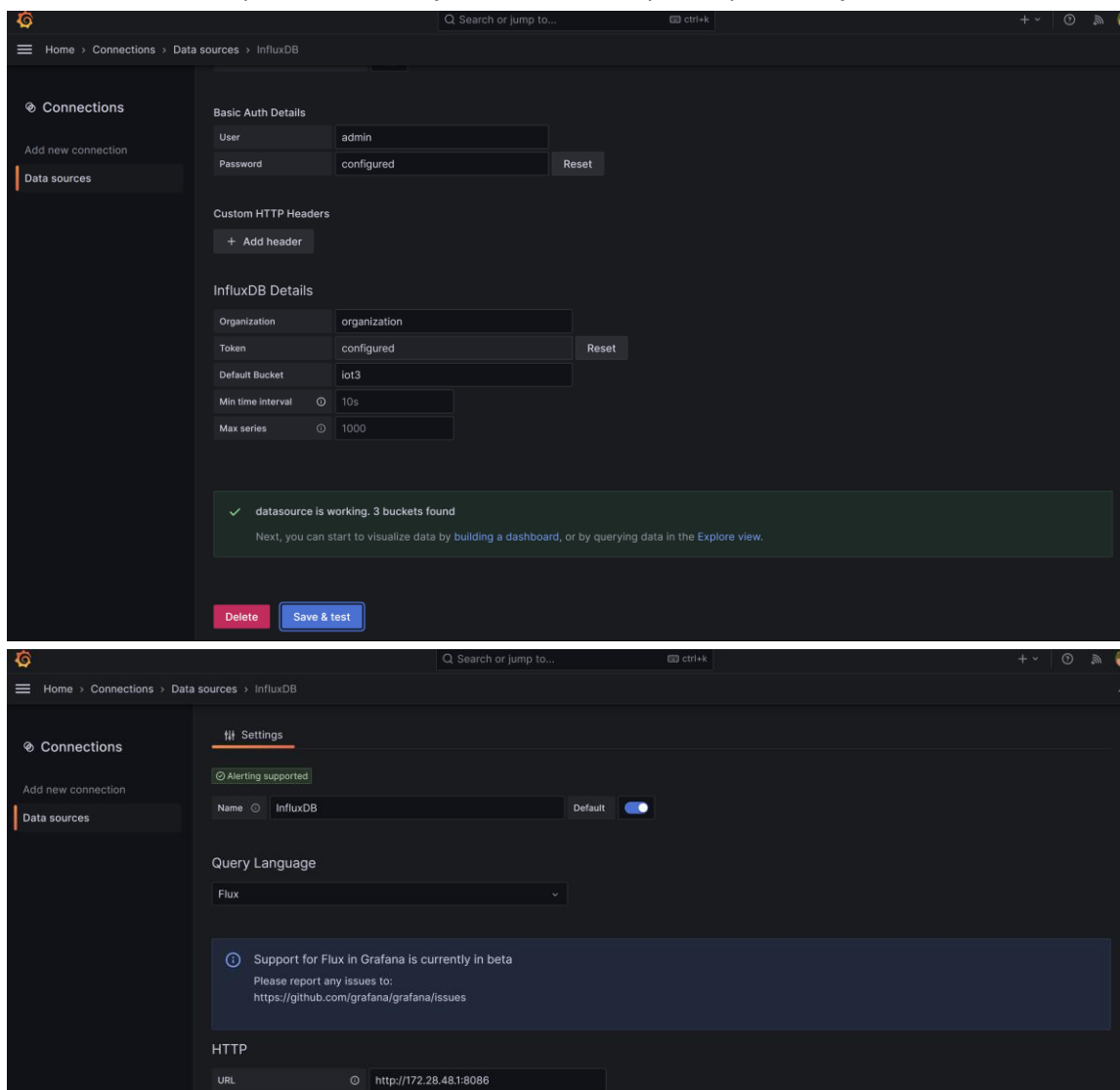
```

    "Temperature sensor",
    "DHT11"
  ],
  "location": "Nis",
  "service": {
    "name": "edgex-device-rest"
  },
  "profile": {
    "name": "SensorCluster2"
  }
}

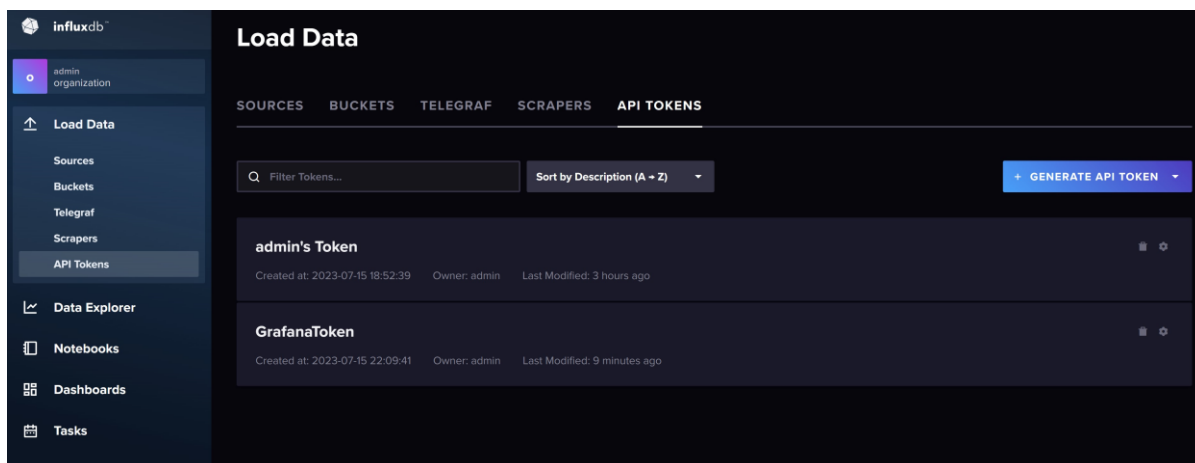
```

InfluxDb i Grafana

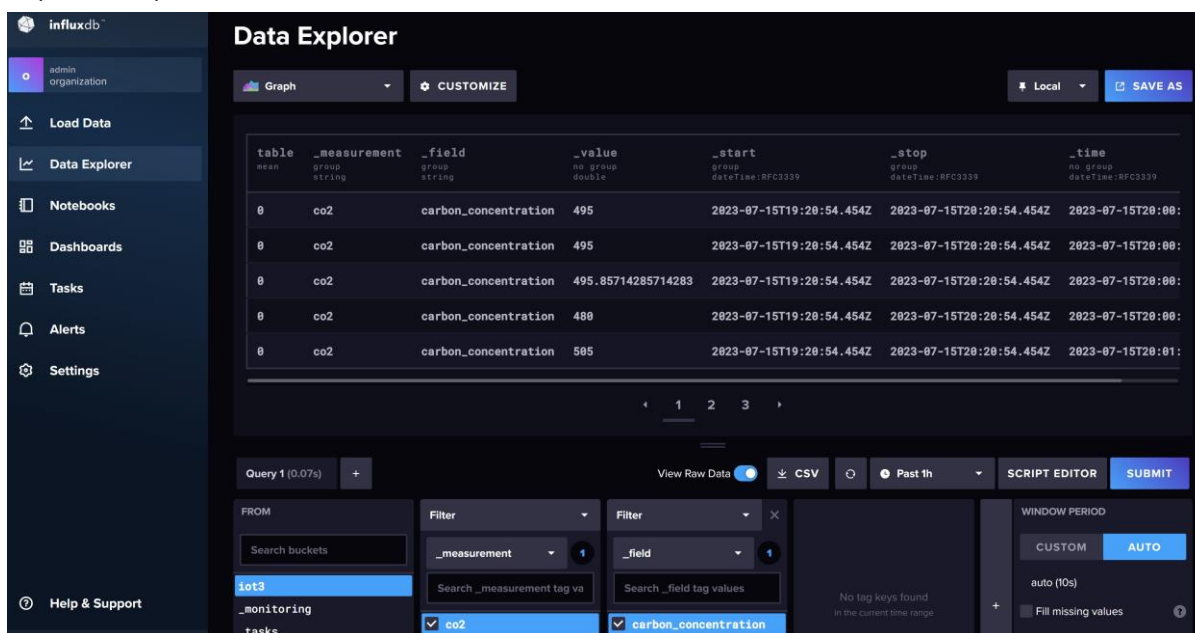
Kada se kreira nalog na InfluxDB-u na portu 8086, treba uneti korisničko ime "admin", lozinku "adminadmin", organizaciju "organization" i bucket "iot3". Nakon toga, može se postaviti Grafana. Početno korisničko ime i lozinka za Grafanu su "admin", uz mogućnost da se kasnije izvrši promena. Za Grafanu, treba uspostaviti konekciju sa InfluxDB-om prema prikazanoj slici.



U cilju dobijanja API tokena, potrebno je otići u InfluxDB i generisati novi All Access Token.

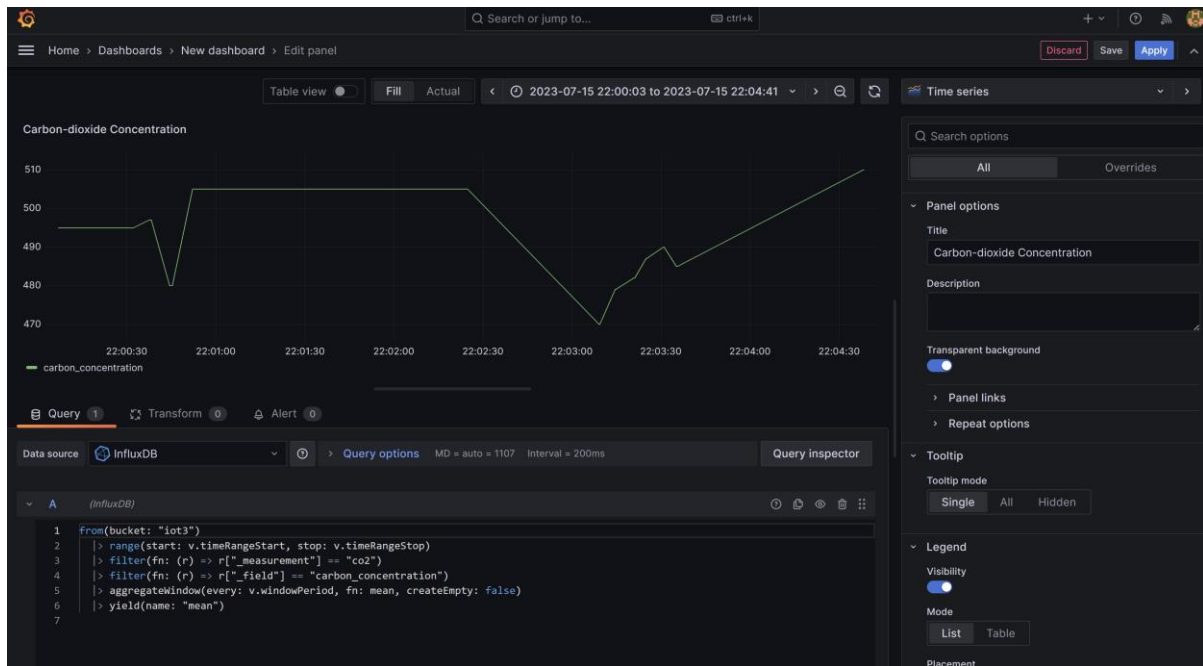


Zatim je potrebno napraviti Dashboard koji će vizualizovati podatke iz selektovane konekcije. Nakon toga, potrebno je otići u InfluxDB i selektovati podatke koje želimo da vizualizujemo, i kopirati skriptu u Grafani.



Vizuelizacija podataka prikazana je na slici u nastavku.





Visualization service

Ovaj servis je pisan u .Net-u. Cilj mu je da prikupi podatke sa MQTT brokera, nakon čega će oni biti smešteni u InfluxDB bazu podataka, gde će se, uz pomoć Grafane, izvršiti njihova vizuelizacija.

Monitoring service

Ovaj servis pisan je u node.js-u. Ima za cilj da skupi podatke koje EdgeX šalje na MQTT topic, nakon čega se ti podaci prosledjuju eKuiper servisu, koji detektuje koncentraciju ugljen-dioksida (co2) van opsega, nakon čega obaveštava Device preko EdgeX-a o potrebnim akcijama koje treba preduzeti.

```
if (co2 < 490 && currentState === 'OFF')
{
    currentState = 'ON'
    console.log('CURRENT STATE CHANGED TO ON')
    sendAlert()
    return
}

if (co2 > 500 && currentState === 'ON')
{
    currentState = 'OFF'
    console.log('CURRENT STATE CHANGED TO OFF')
    sendAlert()
}
```

Pre pokretanja ovog servisa, potrebno je pokrenuti jednostavan interfejs color changer. Interfejs color changer pokrece se sledećim komandama:

```
docker build -t colorChanger .
docker run -d -p 5000:5000 --name colorchanger colorchanger:latest
```

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 sistema

