

Za dataset je iskoriscen fish pond koji se nalazi u folderu sensorData. U okviru istog foldera se nalazi i jednostavna python skripta koja služi za slanje podataka edgex-u.

Za MQTT broker koriscen je lokalni Eclipse Mosquitto broker, koji se izvrsava na portu 1883, zbog toga je potrebno u docker-compose fajlu promeniti njegovu lokalnu adresu.

EdgeX

Za postavljanje EdgeX-a potrebno je da se konfiguriše izgled podataka koji dolaze, Device Profile i Device.

Konfigurisanje izgleda podataka

Kroz Postman ili neki drugi interfejs potrebno je poslati POST requeste ka http://<edgex_ip>:48080/api/v1/valuedescriptor, sa sledecim podacima u body:

```
Body1: {
  "name": "temperature",
  "description": "Sensor cluster temperature values",
  "min": "-128",
  "max": "100",
  "type": "Int64",
  "defaultValue": "0",
  "formatting": "%s",
  "labels": [
    "temperature"
  ]
}

Body2: {
  "name": "oxygen",
  "description": "Sensor cluster dissolved_oxygen values",
  "min": "0",
  "max": "100",
  "type": "Float64",
  "defaultValue": "0",
  "formatting": "%s",
  "labels": [
    "oxygen"
  ]
}

Body3: {
  "name": "pH",
  "description": "Sensor cluster pH values",
  "min": "0",
  "max": "12",
```

```

        "type": "Float32",
        "defaultValue": "0",
        "formatting": "%s",
        "labels": [
            "pH"
        ]
    }

    Body4: {
        "name": "ammonia",
        "description": "Sensor cluster ammonia values",
        "min": "0",
        "type": "Float32",
        "defaultValue": "0",
        "formatting": "%s",
        "labels": [
            "ammonia"
        ]
    }

    Body5: {
        "name": "nitrate",
        "description": "Sensor cluster nitrate values",
        "min": "0",
        "type": "Int64",
        "defaultValue": "0",
        "formatting": "%s",
        "labels": [
            "nitrate"
        ]
    }
}

```

Postavljanje device profile

Na adresu http://<edgex_ip>:48081/api/v1/deviceprofile/uploadfile u okviru POST request-a, poslati fajl, sa ključem “file”, “device profile.yml”.

Registrovanje device-a

Na adresu http://<edgex_ip>:48081/api/v1/device poslati POST request sa sledecim podacima u body:

```

{
  "name": "Fish_pond_sensor_cluster_03",
  "description": "Fish pond state",
  "adminState": "unlocked",
  "operatingState": "enabled",
  "protocols": {

```

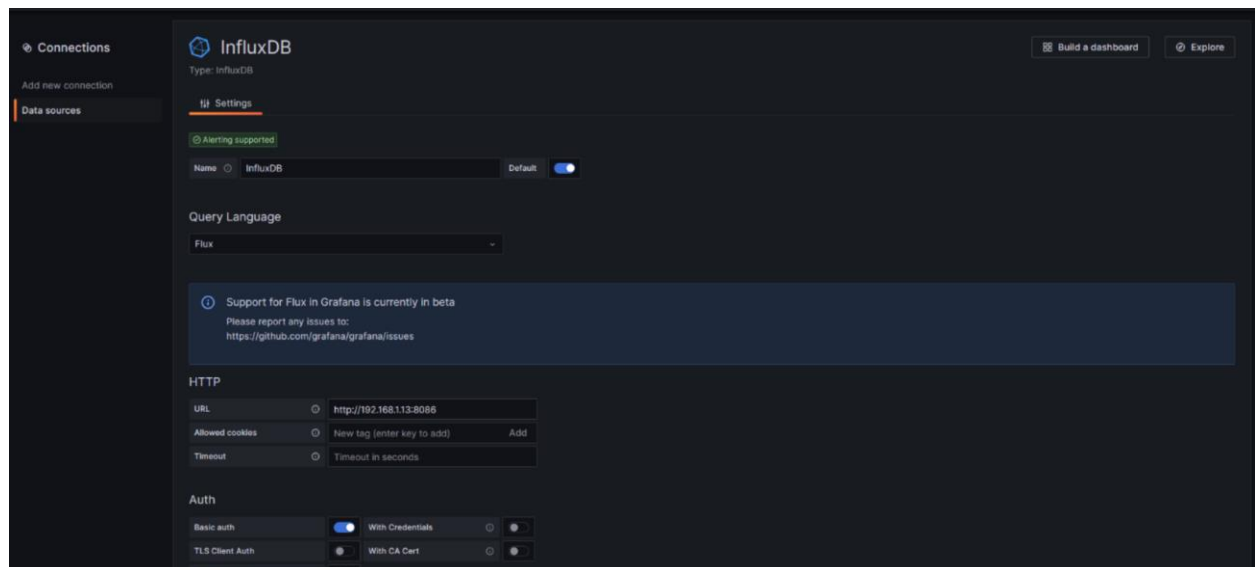
```

    "example": {
      "host": "dummy",
      "port": "1234",
      "unitID": "1"
    },
    },
    "labels": [
      "temperature",
      "pH",
      "ammonia",
      "nitrate",
      "oxygen"
    ],
    "service": {
      "name": "edgex-device-rest"
    },
    "profile": {
      "name": "FishPondSensorCluster"
    }
  }
}

```

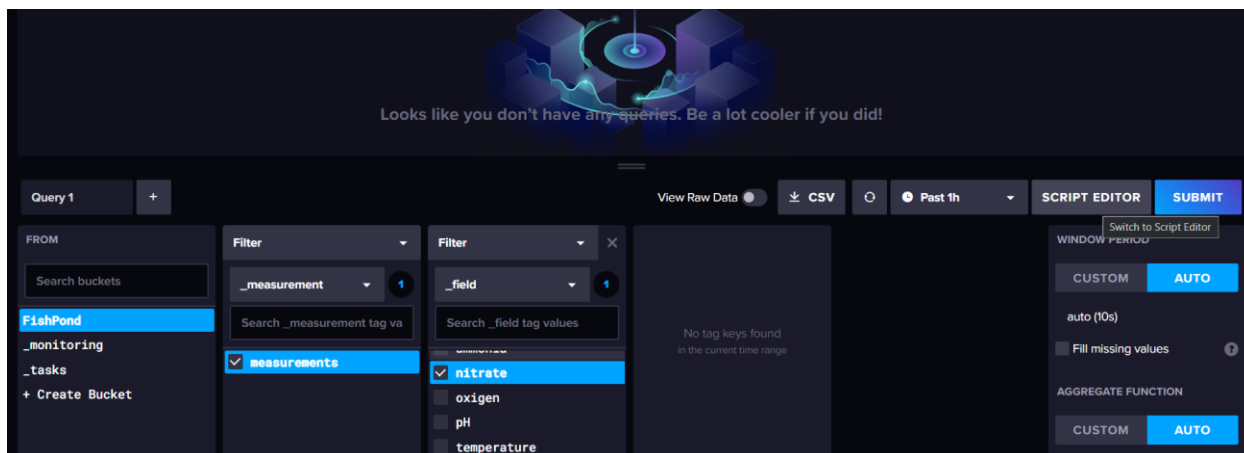
InfluxDb i Grafana

Pri kreiranju naloga na InfluxDB-u na portu 8086, potrebno je uneti username “root”, password “password”, organizaciju Jojobi i bukcet FishPond. Nakon toga potrebno je postaviti grafanu. Inicijalni username i password za grafanu je admin. Kod Grafane, potrebno je napraviti konekciju sa InfluxDB kao sa slike.

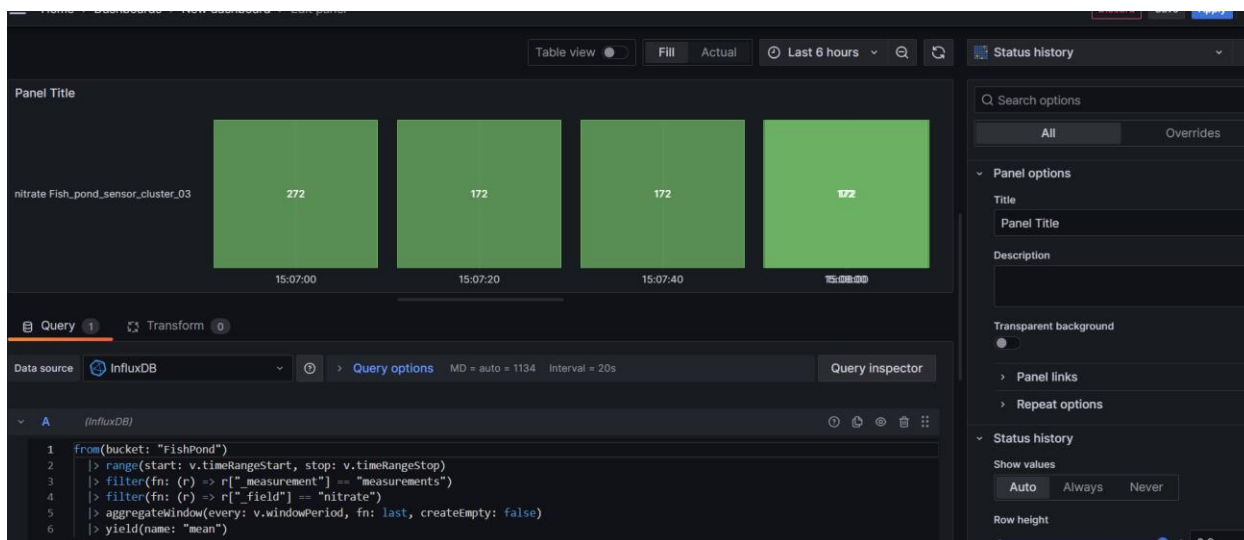


Kod URL-a je potrebno staviti IP adresu mašine na kojoj su pokrenuti docker kontejneri.

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, kliknuti na. SCRIPT EDITOR i kopirati skriptu u Grafani.



Nakon toga treba da se vidi ovako nešto na ekranu.



Visualization service

Ovaj servis je pisan u python-u. Cilj mu je da prikupi podatke sa MQTT brokera, i da ih smesti u InfluxDB, gde će se korišćenjem Grafane vizualizovati podaci. Pre pokretanja potrebno je napraviti .env fajl i unutar njega postaviti potrebne promenljive, koje su navedene u .env.example fajlu.

Monitoring service

Monitoring service je pisan u node.js-u. Ima za cilj da skupi podatke koje EdgeX šalje na MQTT topic, zatim da te podatke salje eKuiper servisu koji detektuje situaciju ukoliko je koncentracija nitrata u vodi van opsega, i zatim obavestava Device preko EdgeX-a o potrebnim akcijama koje treba da preduzme.

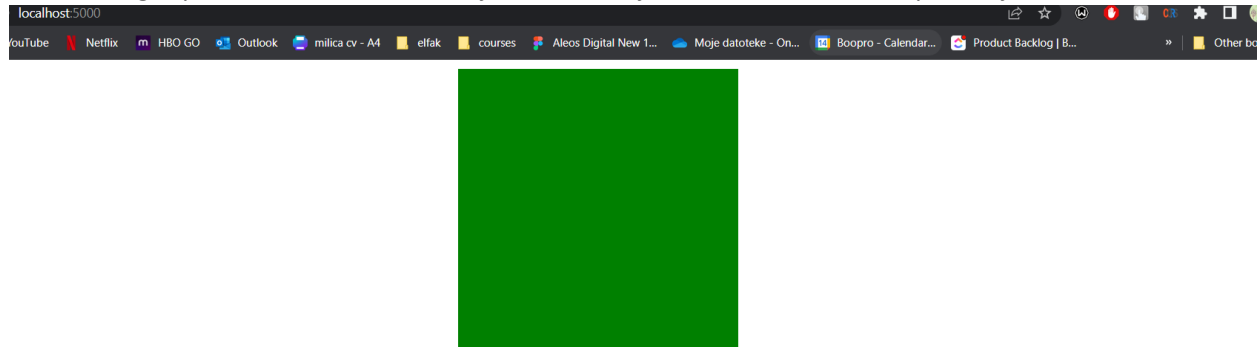
Pre pokretanja ovog servisa potrebno je pokrenuti jednostavan interfejs color changer. Pokrece se komandama:

```
docker build -t colorChanger .
```

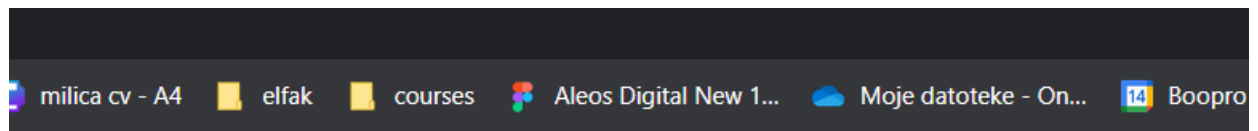
```
docker run -d -p 5000:5000 --name colorchanger colorchanger:latest
```

A zatim pokrenuti skriptu za kreiranje REST device. Skripta kao I yml fajl se nalaze u folderu monitoring.

Color changer prima komande. Ukoliko je koncentracija nitrata dobra onda s prikazuje zeleni kvadrat

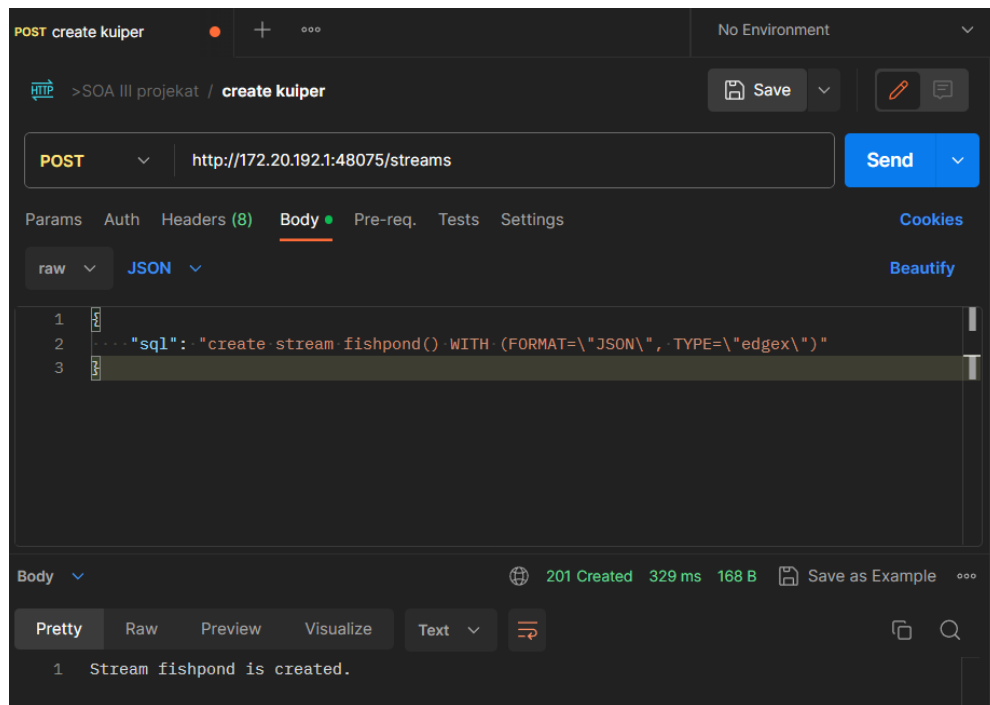


Ukoliko nije onda se prikazuje crveni

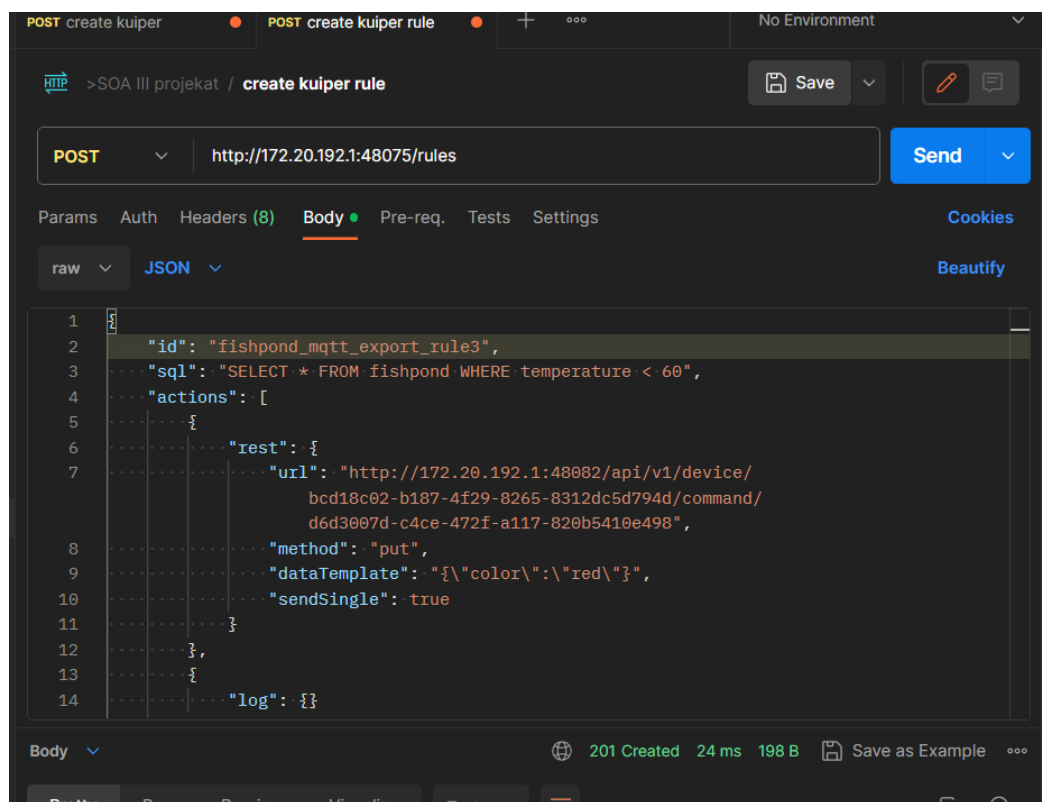


eKuiper

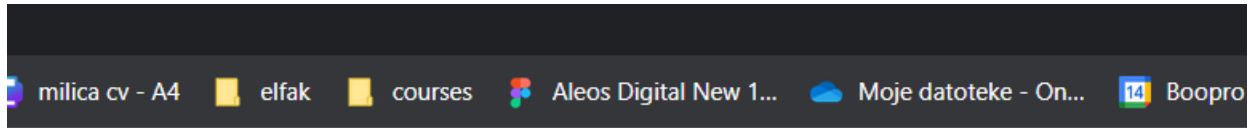
Prvo kreiramo stream sa nazivom fishpond



Zatim kreiramo pravila. Slacemo komandu uredjaju ako je temperatura manja od 60.



Komandu saljemo na isti device kao I kod monitoringa. Ukoliko je temperature manja od 60 pojave se crveni kvadrat



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 generateData.py` iz `sensorData` foldera.