

# Projekat 2 – Uputstvo

## Sensor Dummy service

SensorDummy service je napisan u Node.js-u. Koristi dodatne npm pakete za čitanje iz CSV fajla i za konektovanje i rad nad MQTT brokerom. Prilikom pisanja docker compose file-a, mora se voditi računa da ovaj kontejner krene za izvršavanjem tek nakon što se EMQX broker startuje. On podatke pročitane iz .csv fajla pablišuje na topic sensor\_dummy/values.

## Analytics service

Analytics service je napisan u .Net-u. Koristi dodatne pakete za rad sa MQTT-om i sa InfluxDB-jem. To su paketi MQTTnet i InfluxDB.Client. Za njega isto važi kao i za Sensor Dummy service, treba biti pokrenut tek nakon što je EMQX startovan. Čita podatke sa sensor\_dummy/values topic-a, zatim ih pablišuje na analytics/values topic eKuiper-u. Obrađene podatke eKuiper-a prima sa eKuiper/anomalies topic-a, i upisuje ih u InfluxDB.

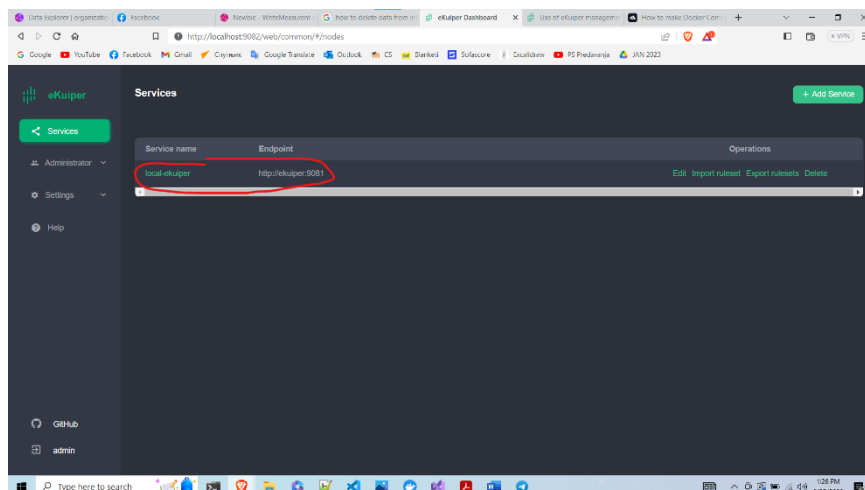
## MQTT broker

Mqtt broker koji sam koristio je EMQX, zato što sam, pre početka rada na projektu, video da se preporučuje da se za eKuiper koristi EMQX broker.

## eKuiper service

U docker compose file-u sam stavio da je MQTT\_SOURCE\_\_DEFAULT\_\_SERVER: "tcp://192.168.100.6:1883", jer je to adresa mog računara. Nije htelo da se kači na mqtt broker ukoliko navedem localhost za adresu. Pokrećem i manager container za eKuiper, kako bih lakše, preko UI-a postavio source i rule za svrhu projekta.

Kada se pokrene kontejner, potrebno je otići na adresu localhost:9082, i logovati se korišćenjem



username: admin, password: public. Nakon toga, potrebno je kliknuti na service koji nam se prikazao.

Zatim je potrebno ići na stream karticu, odabrati Create Stream dugme i uneti sledeće podatke sa slike, a potom kliknuti dugme Submit.

The screenshot shows the 'Create Stream' form in the eKuiper Dashboard. The form fields are as follows:

- Stream Name:** `iot2stream`
- Whether the schema stream:** ☐
- Stream Type:** `mqtt`
- Data Source (MQTT Topic):** `analytics/values`
- Configuration key:** `Select`
- Stream Format:** `json`
- Shared:** ☒ true ☐ false

At the bottom of the form are **Submit** and **Cancel** buttons. The left sidebar shows the navigation menu with 'Services' selected.

Zatim je potrebno otići na Rules karticu kako bi se unelo pravilo po kome se „filtriraju“ podaci sa prethodno definisanog stream-a. Potrebno je otići na Create Rule dugme, a zatim uneti sledeće podatke sa slike.

The screenshot shows the 'Create Rule' form in the eKuiper Dashboard. The form fields are as follows:

- Rule ID:** `eKuiperRule`
- Name:** `Rule to issue alarm when temperature inside the room is above 29.5 degrees`
- SQL:**

```
1 SELECT *
2 FROM iot2stream
3 WHERE temp > 29.5 AND out_in = "In"
```

Below the SQL field is the **Actions** section, which includes a table with the following data:

Sink	Operations
<code>mqtt</code>	<code>Edit</code> <code>Delete</code>

At the bottom of the form is an **Options** section. The left sidebar shows the navigation menu with 'Rules' selected.

Zatim je potrebno dodati akciju na dugme +Add sa sledećim podacima.

\* Sink Documentation

mqtt

Resource ID + Add sink template

Select

MQTT broker address ? MQTT topic ?

tcp://192.168.100.6:1883 eKuiper/anomalies

MQTT ClientID ? MQTT protocol version ?

MQTT ClientID MQTT protocol version

3.1.1

QoS ? Username ?

QoS Username

Password ? Certification path ?

Password Certification path

Private key path ? Root Ca path ?

Private key path Root Ca path

Skip Certification verification ?

☐ True ☐ False

Omit if content is empty ? Send single ?

☐ True ☒ False ☒ True ☐ False

Stream Format

json

Data template ?

Advanced

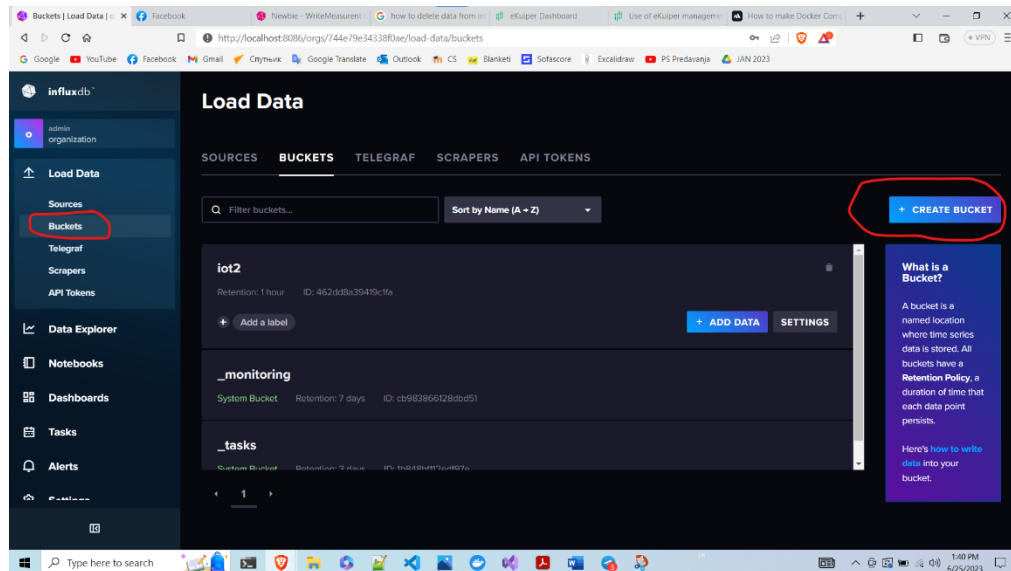
Cancel Test Connection

Nakon toga je potrebno kliknuti Submit dugme, i startovati Rule.

Za njega važi isto kao i za prethodna 2 servisa, mora sačekati EMQX broker da se podigne, kako bi se on sam pokrenuo.

## InfluxDB

Nakon pokretanja, potrebno je logovati se na InfluxDB kreiranjem naloga i potrebno je da se kreira Bucket, kako bismo imali gde da smeštamo podatke. Prilikom kreiranja Bucket-a, ja sam stavio da se podaci brišu nakon jednog sata, kako mi se ne bi duplirali podaci nakon nekog vremena korišćenja.



Za kraj ću prikazati celu arhitekturu(tok dešavanja) u aplikaciji.

