

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 sa izvršavanjem tek nakon što se EMQX broker startuje. On podatke pročitane iz .csv fajla publish-uje na topic sensor_dummy/values.

Analytics service

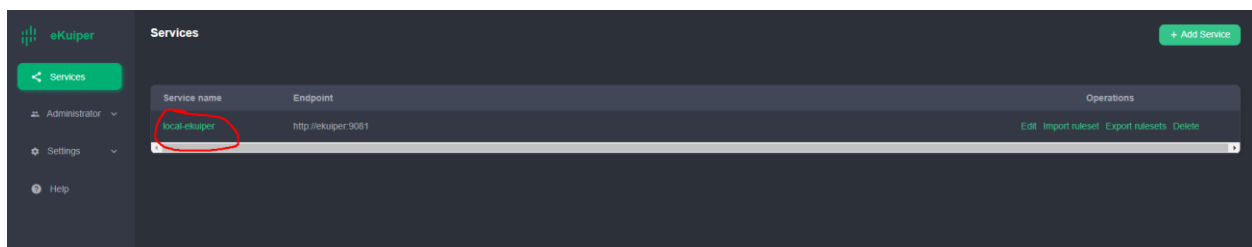
Analytics service je napisan u .Net-u. Koristi dodatne pakete za rad sa MQTT-om i sa InfluxDB-jem. To su paketi MQTTnet i InfluxDB.Client. Za njega isto važi kao i za Sensor Dummy Service, treba biti pokrenut tek nakon što je EMQX startovan. Čita podatke sa sensor_dummy/values topic-a, zatim ih publish-uje 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 je korišćen je EMQX, zato što se preporučuje da se za eKuiper koristi EMQX broker.

eKuiper service

U docker compose file-u je stavljeno da je MQTT_SOURCE__DEFAULT__SERVER: "tcp://emqx:1883", gde je emqx naziv kontejnera EMQX brokera. Pokreće se i manager kontejner za eKuiper, kako bi se lakše, preko UI-a postavio source i rule za svrhu projekta. Kada se pokrene kontejner, potrebno je otići na adresu localhost:9082, i ulogovati se korišćenjem username: admin, password: public. Nakon toga, potrebno je kliknuti na service koji nam se prikazao.



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

Source / Edit

Stream Name:

☐ Whether the schema stream

Stream Type:

Data Source (MQTT Topic):

Configuration key: [+ Add configuration key](#)

Stream Format:

Shared: ☐ true ☒ false

[Submit](#) [Cancel](#)

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

Rules / Edit

Rule ID:

Name:

SQL:

```
1 SELECT *
2 FROM iot2stream
3 WHERE Temperature > 20 AND Humidity < 30
```

Actions [+ Add](#)

Sink	Operations
mqtt	Edit Delete

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

Edit action

✕

* Sink

Documentation

mqtt

Resource ID

+ Add sink template

Select

* MQTT broker address

?

tcp://emqx:1883

* MQTT topic

?

eKuiper/anomalies

MQTT ClientID

?

MQTT protocol version

?

3.1.1

QoS

?

Username

?

Password

?

Certification path

?

Private key path

?

Root Ca path

?

Skip Certification verification

?

True

False

Omit if content is empty

?

True

False

Send single

?

True

False

Stream Format

json

Data template

?

Advanced

Cancel

Submit

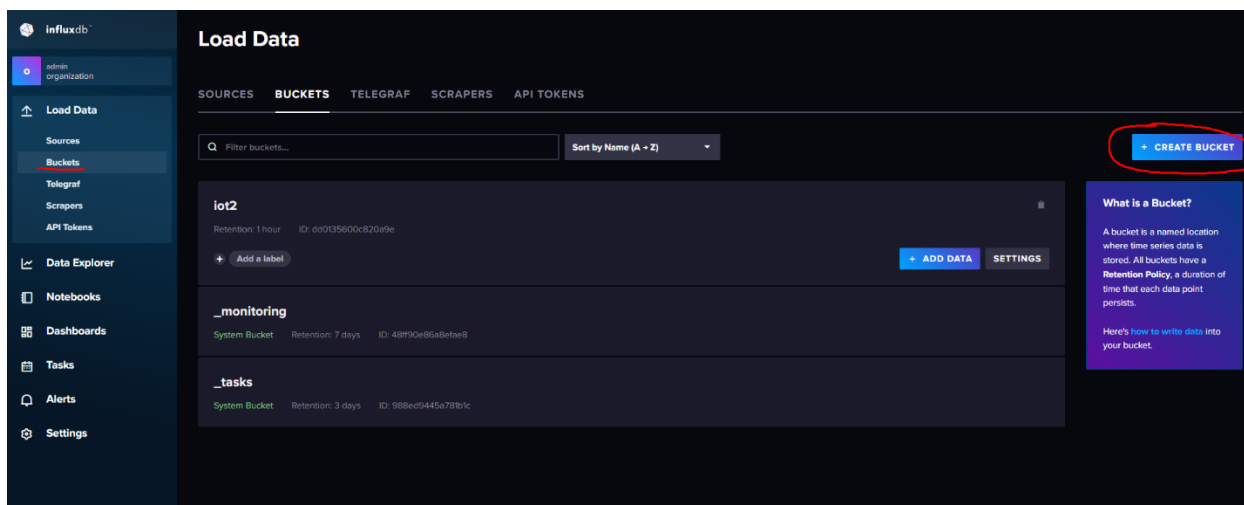
Test Connection

Nakon toga je potrebno kliknuti na 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 i on sam pokrenuo.

InfluxDB

Nakon pokretanja, potrebno je ulogovati se na InfluxDB kreiranjem naloga sa username-om:admin i password-om:adminadmin. Zatim potrebno je da se kreira Bucket pod nazivom `iot2`, kako bismo imali gde da smeštamo podatke. Prilikom kreiranja Bucket-a, stavljeno je da se podaci brišu nakon jednog sata, kako se ne bi duplirali podaci nakon nekog vremena korišćenja.



Izgled arhitekture(tok dešavanja) u aplikaciji:

