**Servisno-orijentisanje arhitekture – Projekat 2**

Cilj ovog projekta je da omogući studentima priliku za sticanje praktičnog iskustva u razvoju i integraciji mikroservisa korišćenjem različitih popularnih tehnologija. Ovaj projekat će demonstrirati implementaciju mikroservisne arhitekture u različitim tehnološkim okruženjima, uključujući .NET, Node.js, Java/Spring Boot i PythonFlask .

U projektu je potrebno razviti dva mikroservisa. Prvi mikroservis, SensorDummy, je servis koji simulira očitavanje podataka sa senzora i slanje istih na odredjeni topic preko EMQX brokera. Drugi mikroservis, AnalyticService prima podatke od prvog, analizira ih i smešta u bazu podataka.

SensorDummyService

SensorDummyService je razvijen u NodeJS okruženju. Podatke čita iz .csv fajla i šalje ih preko EMQX brokera na „sensor\_dummy” topic. Za čitanje iz fajla i komunikaciju sa brokerom se koriste dodatni npm paketi. Pokreće se preko docker-compose fajla(komandom docker-compose up -d u cmd-u). Važno je da pre njega bude pokrenut EMQX broker.

AnalyticsService

AnalyticsService je servis razvijen u .NET-u I koristi odgovarajuće NuGet pakete za MQTT broker i InfluxDb bazu podataka. Uloga servisa je da prima podatke preko EMQX brokera sa „sensor\_dummy” topic-a, zatim te podatke šalje prosledjuje eKuiper-u preko “analytics/airpivalues” topica-a, i na kraju podatke koje dobije od eKuiper-a preko "eKuiper/AirQuality" topic-a upisuje u InfluxDB bazu podataka. Servis se takodje pokreće preko docker-compose fajla i važno je da je prvo EMQX broker pokrenut.

eKuiper

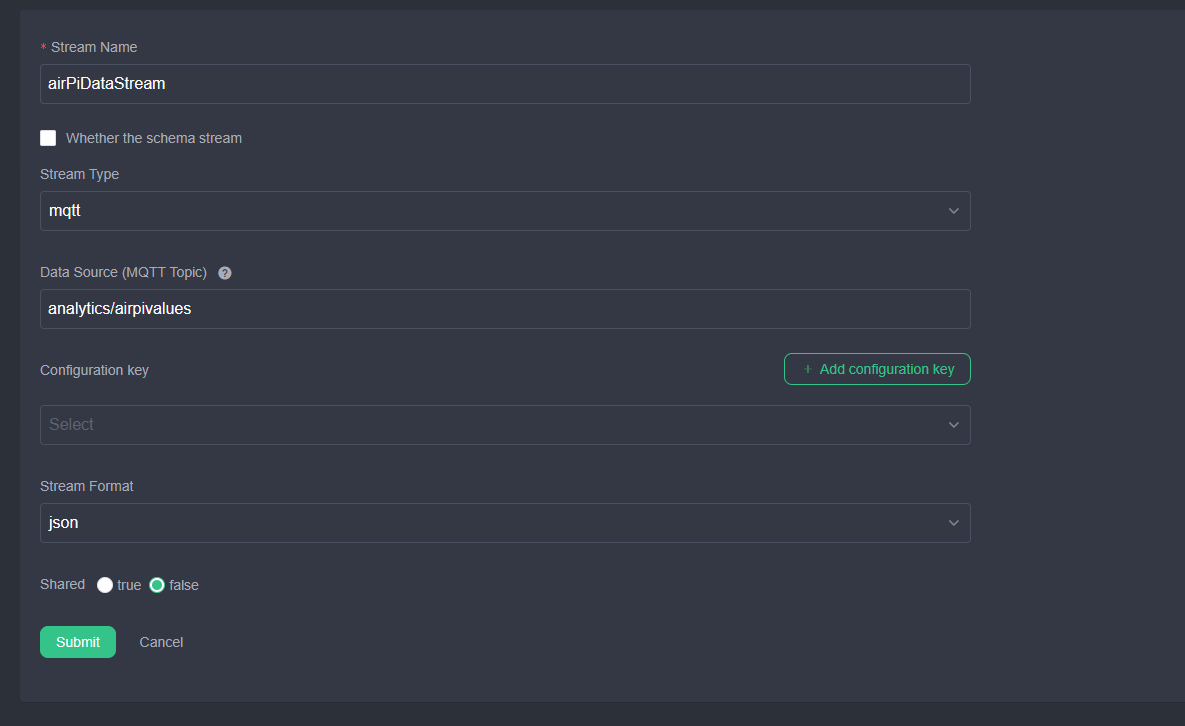
eKuiper je eksterni mikroservis za IoT Stream Processing & Analytics koji prima podatke od Analytics servisa, vrši analizu i vraća podatke po odredjenom pravilu. Pokreće se takodje preko docker-compose fajl, a za njego rad je potrebna odgovarajuća konfiguracija. Konfiguracij se obavlja preko GUI-ja koji nudi management konzola na localhost:9096 adresi.

1. Prvo je potrebno ulogovati se korišćenjem sledećih parametara:

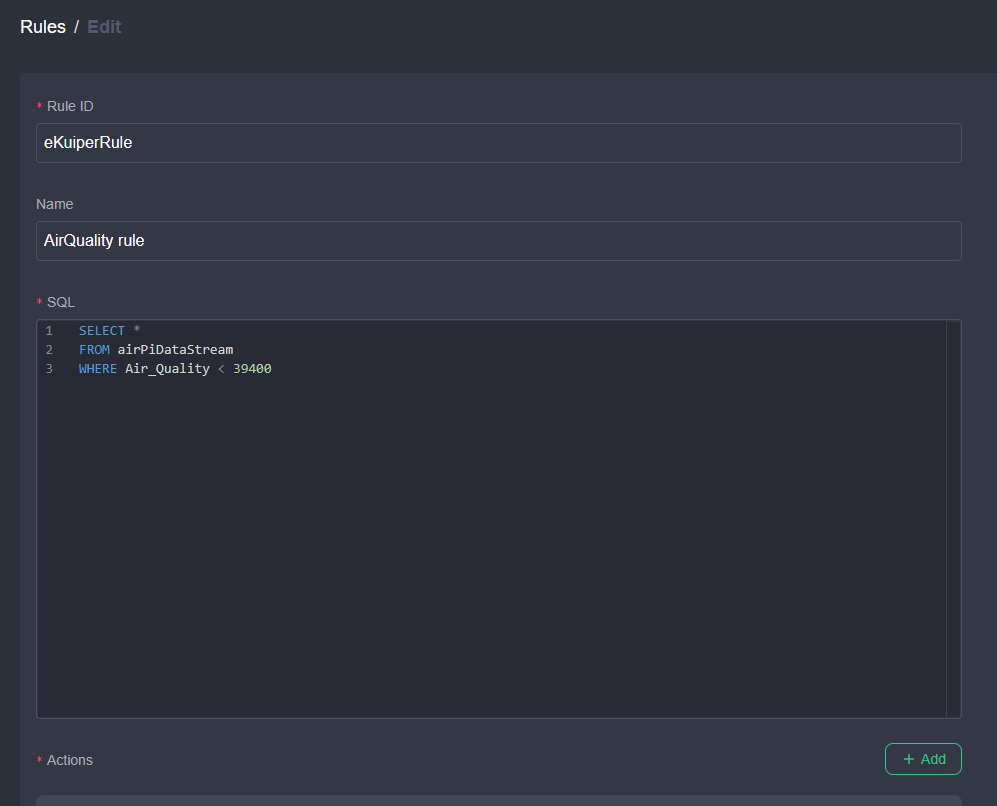
* Username: admin
* Password: public

1. Selektujemo local-ekuiper servis i biramo opciju Create Stream u okviru Source > Stream kartice
2. Unosimo sledeće parametre u okviru Create Stream dijaloga

* Stream Name: airPiDataStream
* Stream Type: mqtt
* Data Source: analytics/values
* Biramo opciju submit



1. Kreiramo pravilo po kome će se filtrirati podaci. U okviru Rules kartice, biramo Create Rule opciju. Unosimo id i ima pravila, a zatim i SQL upit po kome se selektuju podaci iz skupa podataka koji pristižu, po nekom kriterijumu.



1. Na kraju kreiramo odgovarajuću akciju koja će se izvršiti kada naidje podatak koji zadovoljava navedeno pravilo. Izaberemo opciju Add i unesemo sledeće parametre:

* Sink: mqtt
* MQTT broker addres: tcp://172.26.176.1:1883
* MQTT topic: eKuiper/AirQuality
* Ostalo ne menjamo

Testiramo konekciju opcijom Test Connection, a zatim Submit da

kreiramo pravilo

InfluxDB

InfluxDB je baza koju koristimo za time-series podataka koja su ispunila zahteve pravila postavljenih u eKuiper servisu.

Pri inicijalnom pokretanju baze potrebno je uneti sledeće parammetre:

* Username: student
* Password: student2023
* Initial bucket: IoT2
* Organization: AirPI

Nakon ove konfiguracije podatke je moguće videti u okviru Data Explorer kartice.