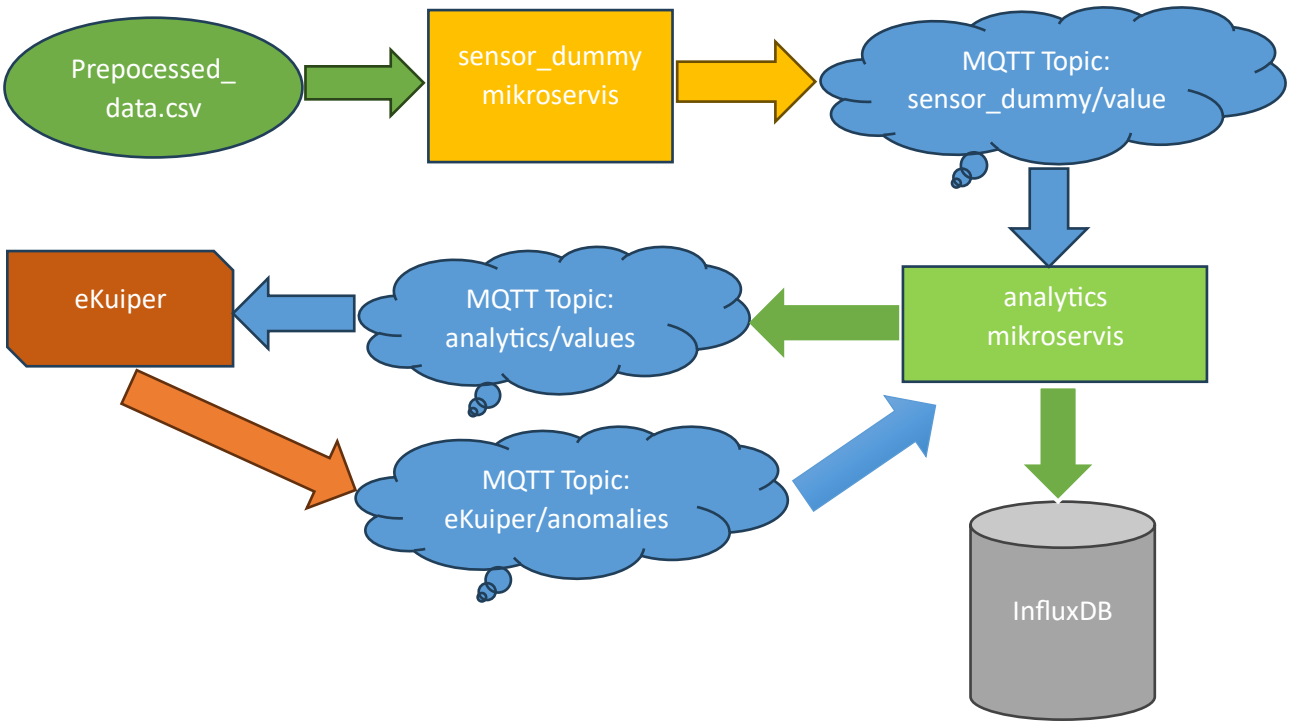
Projekat 2

Cilj ovog projekta je da se implementira mikroservisna arhitektura u razlicitim tehnologijama, kao što su: .NET, NodeJs, Java/Spring Boot, Python/Flask.

Prvi servis koji treba da se razvije je Sensor Dummy i on treba da sa senzora čita podatke i da te podatke šalje na određeni topic MQTT brokera.
Drugi servis koji treba da se implementira nazvan je Analytics, on treba da se pretplati na MQTT message broker i da dobije podatke koji su emitovani na topic od strane prvog servisa. Podakte koje dobija treba da se smeštaju u NoSQL bazu podataka InfluxDB.

Arhitektura sistema



Sensor Dummy mikroservis

Sensor Dummy je implementiran u programskom jeziku JavaScritp i koristi Node.js okruženje. Potreban je dodatan paket za rad sa MQTT i za čitanje podataka iz CSV fajla. Podatke koje čita ovaj mikroservis šalje ih na topic pod imenom: **sensor_dummy/values**.

Bitno je da se ovaj mikroservis pokrene tek nakon što se pokrene EMQX kako bi podaci mogli da se šalju.

Za MQTT broker je izabran EMQX broker.

Analytics mikroservis

Glavni zadatak ovog mikroservisa jeste da pročita podakte koje Sensor Dummy šalje na topic sa imenom **senor_dummy/values** odgovarajućeg MQTT brokera. Zatim te podatke koje čita upisuje na **analytics/values** topic, preko koga se pročitani podaci šalju eKuiper-u.

17588 Nikola Davinić

Bitno je da se ovaj mikroservis pokrene tek nakon što se pokrene EMQX kako bi mogao da pročita podatke, a i poslao eKupier-u.

Implementiran je u .NET-u verzija 6.0 i koristi dodatne pakete za rad sa MQTT-om i sa InfluxDB bazom podataka.

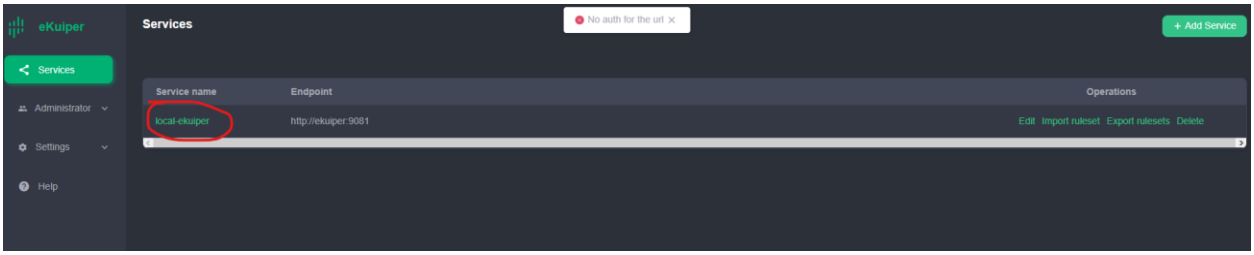
eKuiper

Nakon izvršavanja docker-compose komande potrebno je da se ode na adresu gde se nalazi eKuiper:

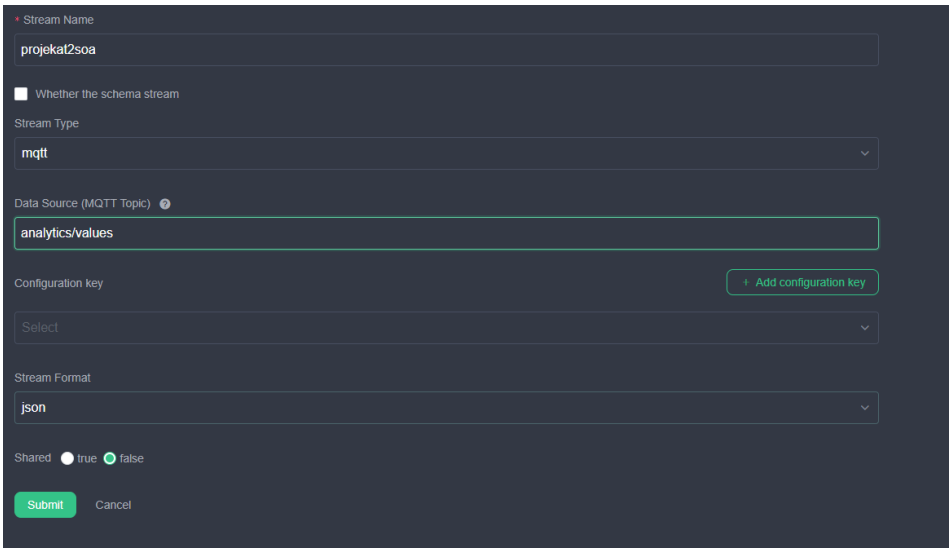
http://localhost:9082

Odlaskom na ovu adresu potrebno izlazi login forma gde je potrebno da se unese sledeće:
Username: admin
Password: public

Nakon što smo ulogovani je potrebno da pritisne na local-ekuiper i tu izvrši dalja konfiguracija kao na slici:

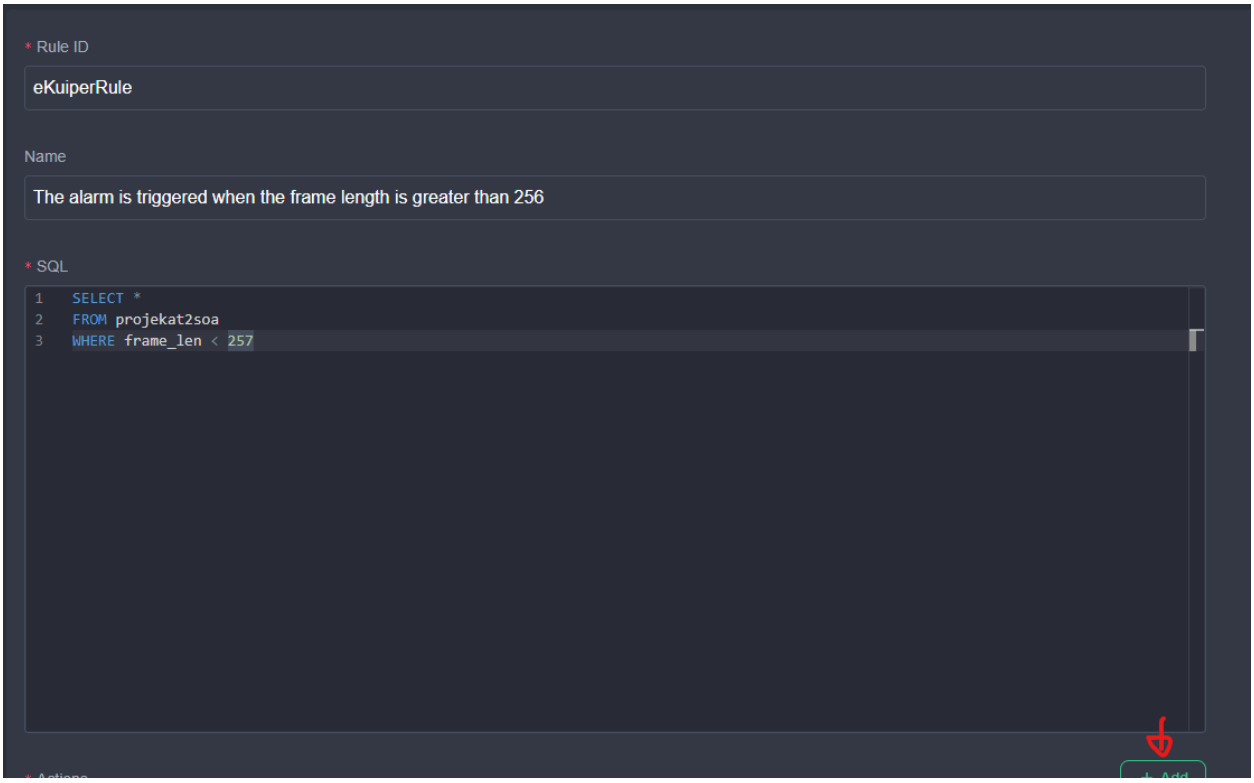


Nakon pritiska na local-ekuiper izlazi deo gde treba da se napravi stream podataka pritiskom na dugme create stream i treba da se popuni sledećim podacima:
Stream name: projekat2soa
Type: mqtt
Data Source: analytics/values (preko kog topic-a dobija podatke)
Ostala polja ostaviti kako je bilo pri otvranju forme.

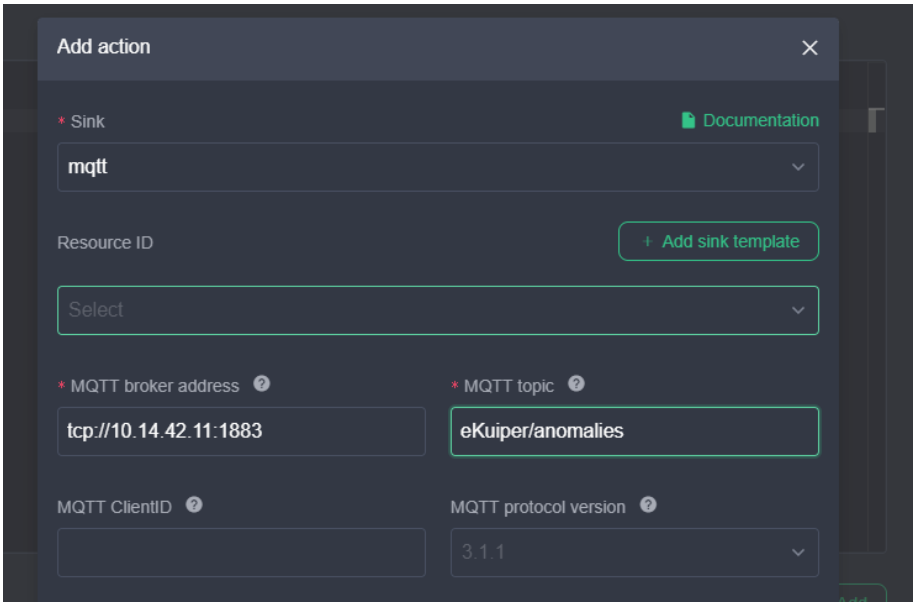


Nakon što se popune podaci popune potrebno je da se to submituje pritiskom na dugme submit.

Sledeći korak je da se napravi pravilo koje će da u zavisnosti od uslova da aktivira alarm (da pošalje podatke ako se ne upali, ili da signalizira ako podaci nisu dobri), to se biranjem kartice sa imenom Rules, pa onda pritiskom na dugme Create Rule. Otvara se dijalog koji treba da se popuni na sledeći način:
Rule ID: eKuiperRule
Name: The alarm is triggered when the frame length is greater than 256
SQL: SELECT *
FROM projekat2soa
WHERE frame_len < 257



Takodje je ovde potrebno da se dalje pritisne na dugme za dodavanje akcije. Popunjavanje dijaloga treba da izgleda na sledeći način:
MQTT topic: eKuiper/anomalies (ovo je ime topic-a gde se šalju podaci i analytics mikroservis je prijavljen na ovaj topic)



Ostale podatke ostaviti kako su i bili prilikom otvaranja dijaloga. Onda pritiskom na dugme Test connection testiramo da li je konekcija dobra i idemo na dugme submit.
Nakon ovog je preostalo da se podesi baza podataka.

InfluxDB konfiguracija

Smeštanje podataka koji su prošli ono pravilo koje je postavljeno kod eKuiper-a vrši se u InfluxDB bazu podataka. Kako bi konfigurisali bazu potrebno je da se uradi sledeće:

1. Da se ode na adresu **http://localhost:8086** i da se kreira nalog na sledeći način:
   Username: admin
   Password: adminadmin
   Organization: organization
   Bucket: iot2
2. Nakon toga je potrebno da se ode na stranicu Data Explorer kako bi se prikazali podaci koji su smešteni u bazi: