

# SPRAWOZDANIE KOŃCOWE

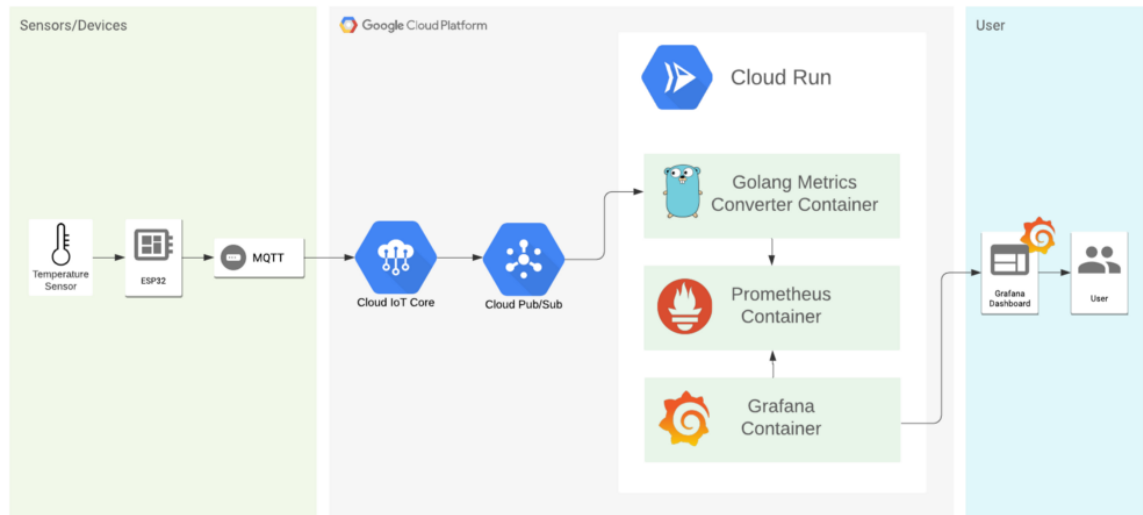
Internet Rzeczy

**Temat:** Wizualizacja danych pomiarowych z czujnika BME280 przy użyciu modułu ESP8266 oraz Google Cloud.

Radosław Siwiec  
Informatyka II stopień,  
stacjonarne,  
2 semestr,  
Gr.1B.

## 1. Założenia ogólne

W projekcie wykorzystano cyfrowy czujnik temperatury, ciśnienia i wilgotności BME280. Układ posiada magistralę I2C i jego podłączenie do modułu ESP8266 jest bardzo łatwe. ESP8266 działa tu jako urządzenie protokołu MQTT i wysyła dane pomiarowe do serwera uruchomionego w Google Cloud. Wizualizacja danych odbywa się przy użyciu oprogramowania Grafana.



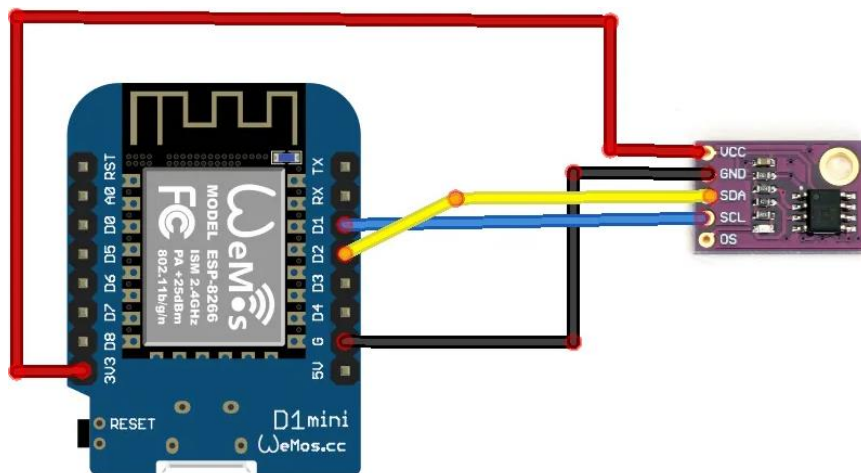
Rys. 1 Schemat ideowy rozwiązania

W projekcie wykorzystano oficjalny poradnik Google Cloud dot. tematyki Internetu Rzeczy dostępny pod adresem:

<https://cloud.google.com/community/tutorials/monitoring-iot-data-grafana>

## 2. Opis implementacji ESP8266

Jak wspomniano wcześniej konfiguracja sprzętowa jest bardzo prosta i nie wymaga użycia żadnych dodatkowych elementów, w tym płytki stykowej. W skład systemu wchodzi jedynie moduł ESP8266 oraz multisensor BME280 podłączony bezpośrednio do modułu ESP odpowiednimi pinami SDA oraz SDL magistrali I2C.



Rys. 2 Schemat połączeń urządzenia

Program sterujący modułu ESP opracowano w środowisku Andino IDE. Program wykorzystuje następujące biblioteki:



Rys. 3 Biblioteki wykorzystane w projekcie

Kod źródłowy projektu znajduje się w zdalnym repozytorium:

<https://github.com/SiwiecSoft/ATH/tree/master/IR/Esp8266-Temp>

### 3. Opis implementacji Google Cloud

Jak wspomniano wcześniej oraz pokazano na rys.1 implementacja rozwiązania chmurowego opiera się na oficjalnym poradniku Google Cloud dostępnym pod adresem internetowym: <https://cloud.google.com/community/tutorials/monitoring-iot-data-grafana> Implementacja jest dość złożona i wieloetapowa dlatego nie zostanie tu powielony ten poradnik, a jedynie pewne kluczowe fragmenty potrzebne do zrozumienia zagadnienia.

Zgodnie z rysunkiem 1 urządzenie ESP wysyła komunikaty w protokole MQTT do serwera Google Cloud gdzie uruchomione są 2 usługi:

- Cloud IoT Core – Zarządzanie urządzeniami IoT podłączonymi do chmury
- Cloud Pub/Sub – broker MQTT, zarządzanie rejestrami urządzeń, subskrypcjami i tematami kanałów MQTT

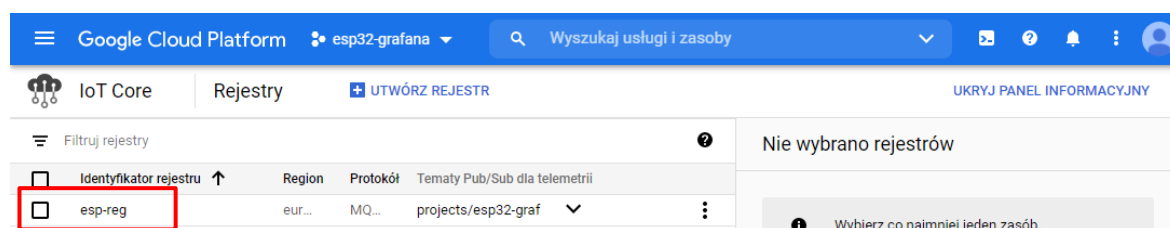
Następnie w usłudze Cloud Run uruchomione są 3 serwisy:

- Konwerter danych z kanału MQTT do kontenera Prometheus
- Server Prometheus – kontener danych
- Server wizualizacji danych Grafana podłączony do kontenera Prometheus

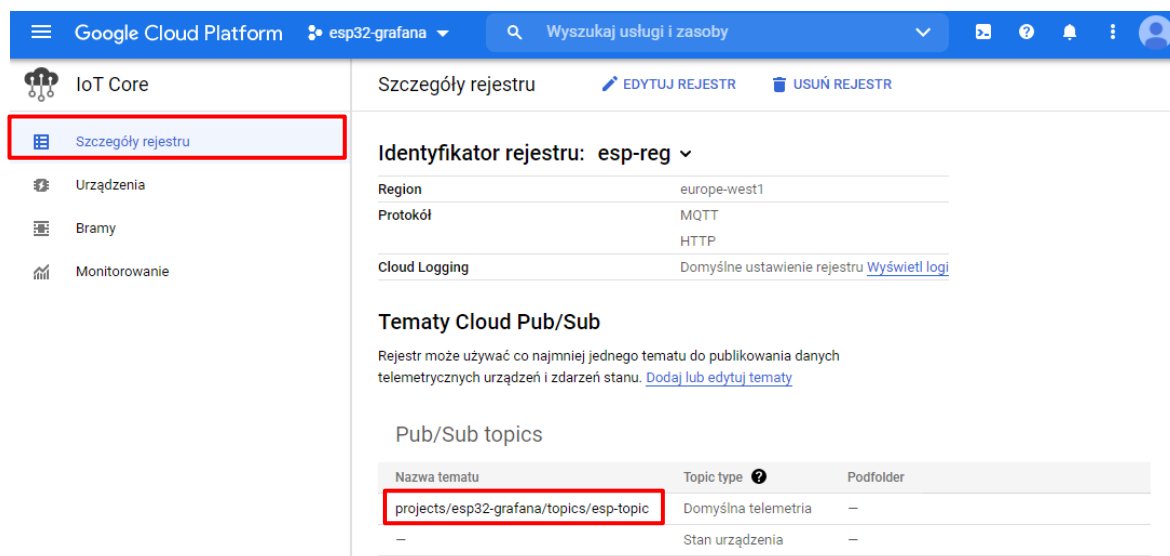
Powyższe serwisy są gotowymi paczkami uruchomianymi w środowisku Docker w usłudze Cloud Run. Ich repozytoria dostępne są pod adresem:

<https://github.com/leozz37/iot-monitoring-gcp-grafana>

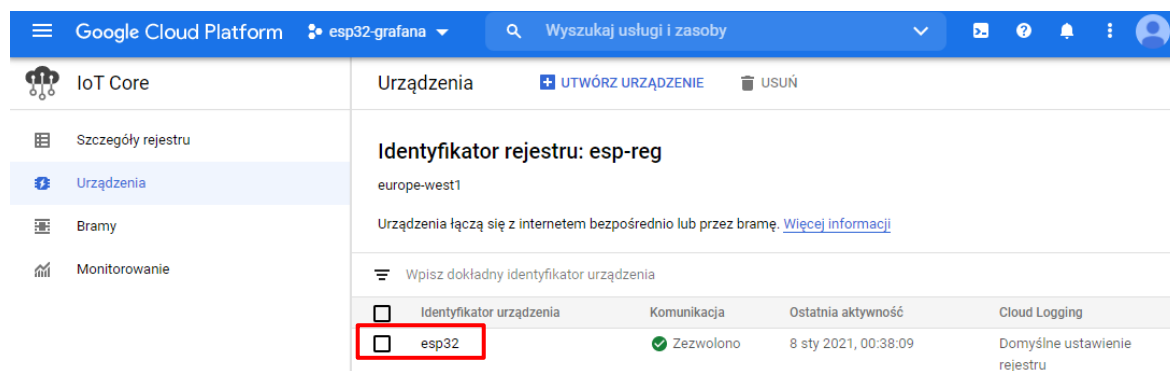
## Konfiguracja rozwiązania w środowisku Google Cloud wygląda następująco:



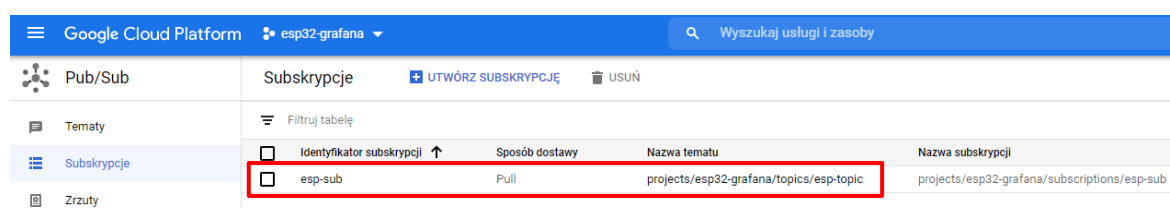
Rys. 4 Usługa IoT Core – rejestry urządzeń – widoczny rejestr esp-reg



Rys. 5 Szczegóły rejestru esp-reg – widoczny temat esp-topic



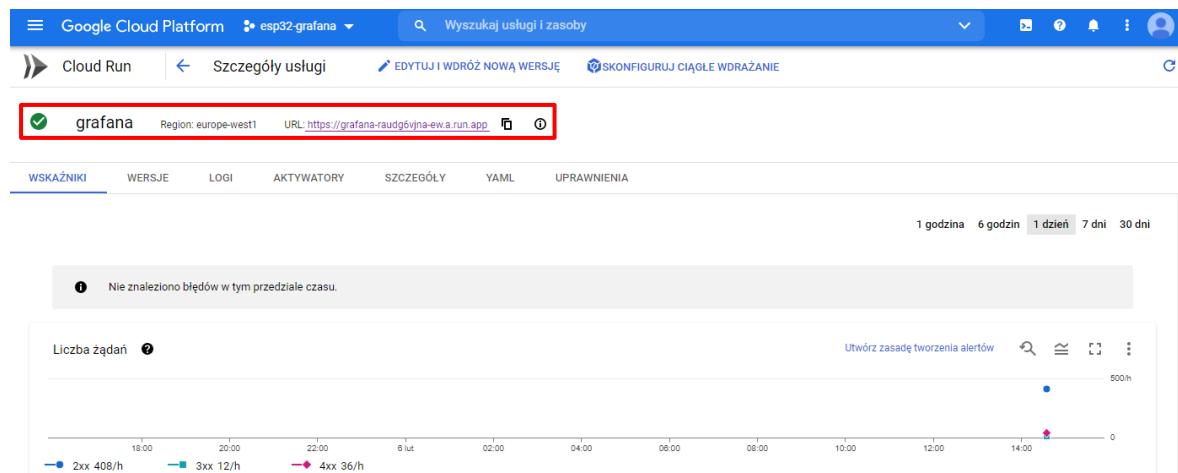
Rys. 6 Urządzenia zarejestrowane w rejestrze esp-reg



Rys. 7 Usługa Pub/Sub – widoczna subskrypcja esp-sub do tematu esp-topic

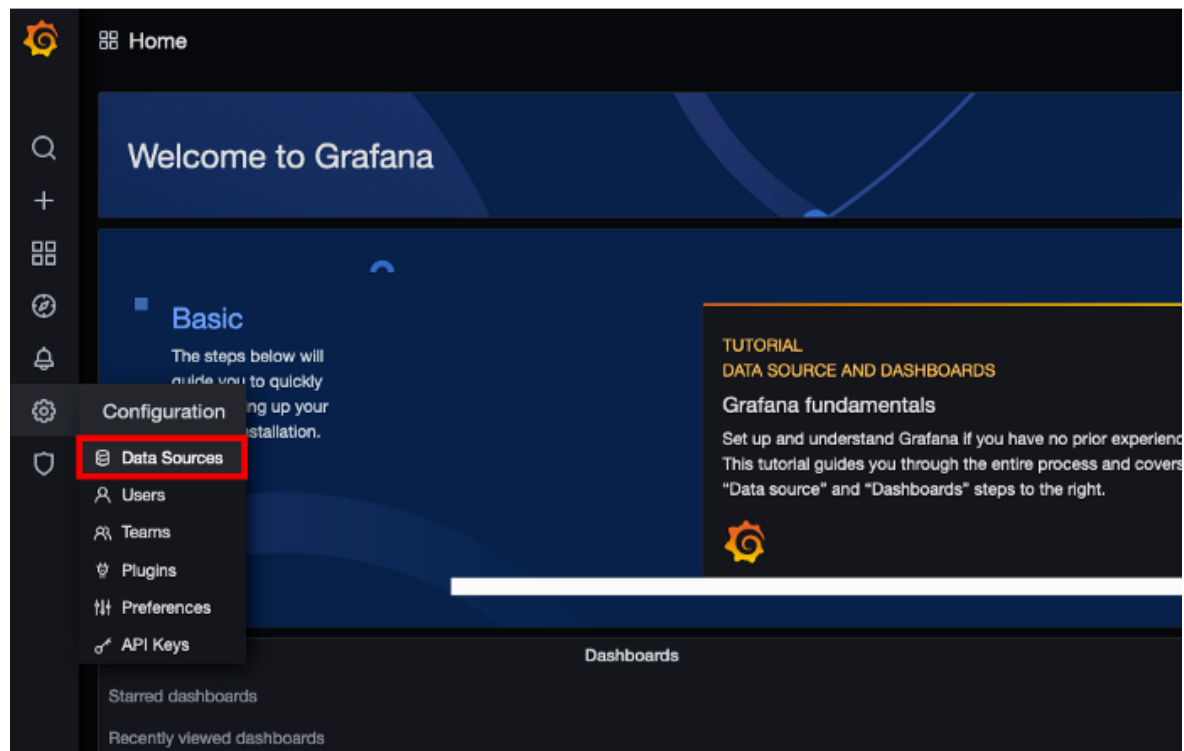
Filtruj	Nazwa	Req/sec	Region	Authentication	Ruch przychodzący	Ostatnio wdrożono	Wdrożenie
<input type="checkbox"/>	golangconv	0	us-central1	Zezwalaj na dostęp nieuwierzytelniony	Wszystkie	7 sty 2021, 22:33:19	srado4ever@gmail.com
<input checked="" type="checkbox"/>	grafana	0	europa-west1	Zezwalaj na dostęp nieuwierzytelniony	Wszystkie	7 sty 2021, 21:24:09	srado4ever@gmail.com
<input checked="" type="checkbox"/>	prometheus	0	europa-west1	Zezwalaj na dostęp nieuwierzytelniony	Wszystkie	7 sty 2021, 21:22:16	srado4ever@gmail.com

Rys. 8 Serwisy uruchomione w usłudze Cloud Run

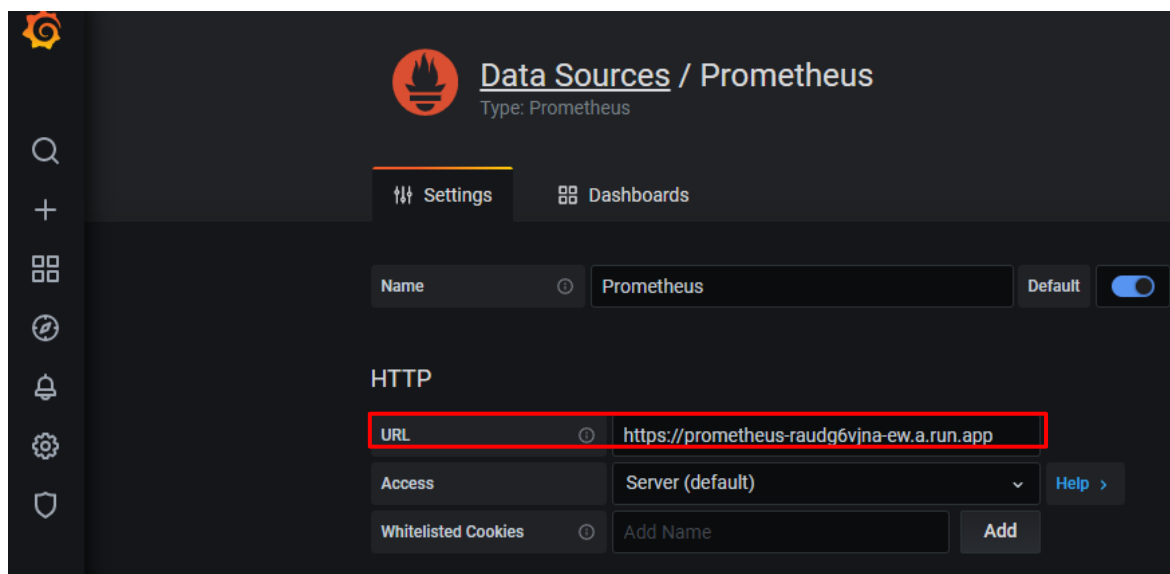


Rys. 9 Szczegóły serwisu grafana w tym adres <https://grafana-raudg6vjna-ew.a.run.app>

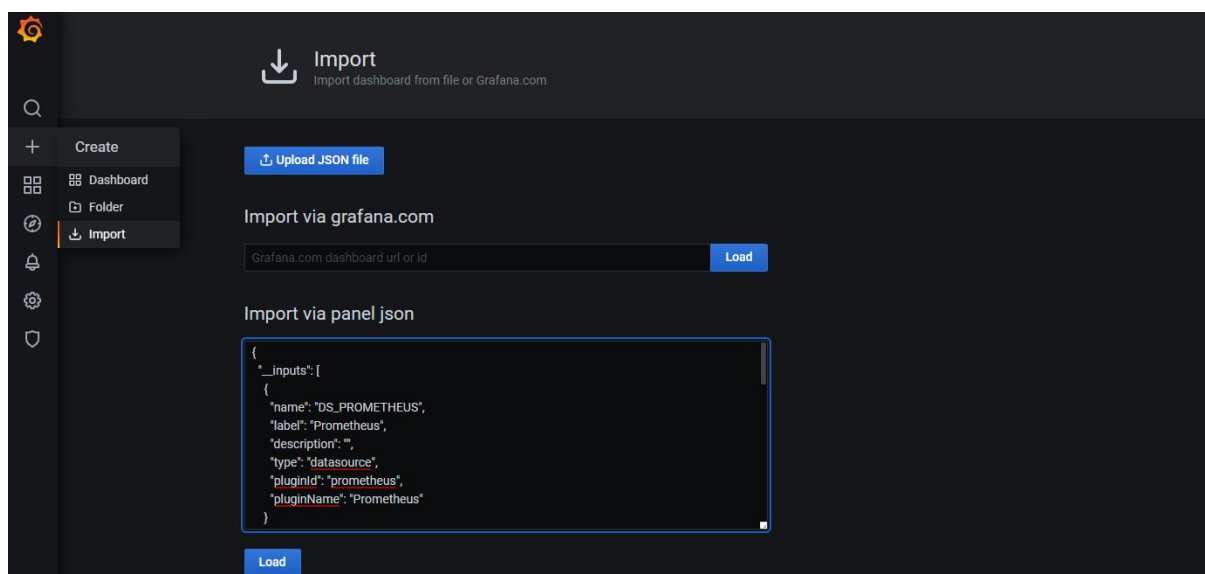
## Uruchomienie serwisu Grafana:



Rys. 10 Dodanie źródła danych



Rys. 11 Ustawienie źródła danych jako kontener Prometheus ze wskazaniem adresu widocznego w szczegółach serwisu prometheus w usłudze Cloud Run



Rys. 12 Zaimportowanie przykładowego dashboard'u w formacie JSON z <https://github.com/leozz37/iot-monitoring-gcp-grafana/blob/master/grafana/grafana.json>



Rys. 13 Dashboard Grafana z widocznymi danymi z czujnika

#### 4. Prezentacja działania aplikacji

Film prezentujący działanie rozwiązania znajduje się w zdalnym repozytorium pod adresem: <https://github.com/SiwiecSoft/ATH/tree/master/IR>

#### 5. Uwagi końcowe

Google Cloud jest potężnym rozwiązaniem chmurowym oferującym szereg możliwości. Istnieje możliwość uruchomienia tu skalowalnego rozwiązania serwerowego które jednego dnia będzie miało kilku użytkowników, a następnego kilka milionów. Developer dostarcza w tym modelu jedynie kod rozwiązania i nie musi się martwić o dostępne zasoby. Opisany projekt udało się zrealizować dzięki próbnej wersji chmury dostępnej za darmo przez 30 dni wraz z pewną sumą do wydania na testowanie płatnych usług.