

Akademia Techniczno-Informatyczna w Naukach Stosowanych

Platformy Chmurowe dla Internetu Rzeczy

Paweł Iljaszewicz, Wrocław 2024

Skrypt przygotowany w ramach projektu „Kształcenie przyszłości: internet rzeczy w zrównoważonej automatyce i robotyce” współfinansowanego ze środków Unii Europejskiej w ramach Programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027, Priorytet 1 Umiejętności, Działanie 01.05 Umiejętności w szkolnictwie wyższym, Typ projektu Dostosowanie oferty podmiotów systemu szkolnictwa wyższego do potrzeb rozwoju gospodarki oraz zielonej i cyfrowej transformacji



Spis treści

1.	Wstęp.....	6
1.1	Internet of Things (IoT),.....	6
1.2	Cloud Computing, czyli obliczeniowa chmura	6
2	Internet rzeczy	11
2.1	Ewolucja od branżowych rozwiązań do globalnej wizji:	12
2.2	Zwiększona integracja danych:	12
2.3	Wzrost znaczenia analizy danych:	12
2.4	Rozwój technologii wspierających IoT:	13
2.5	Wzrost świadomości i akceptacji IoT:	13
2.6	Wyzwania związane z bezpieczeństwem i prywatnością:	13
2.7	Nowe modele biznesowe:.....	13
2.8	Datafication „Datafikacja” to nowa elektryfikacja	14
2.9	Wyzwania w realizacji powszechnego Internetu rzeczy	16
2.9.1	Zarządzanie złożonością	16
2.9.2	Bezpieczeństwo i prywatność danych:	17
2.9.3	Standardy i interoperacyjność:	17
2.9.4	Infrastruktura sieciowa:	18
2.9.5	Koszty wdrożenia:	18
2.9.6	Zarządzanie danymi:	18
2.9.7	Regulacje prawne:	18
2.9.8	Akceptacja społeczna:	19
2.9.9	Złożoność systemów:	19
3	Analiza danych z IoT	19
3.1	Charakterystyka danych IoT:	19
3.1.1	Techniki przetwarzania danych:	19



3.1.2	Analiza eksploracyjna:	20
3.1.3	Uczenie maszynowe i sztuczna inteligencja:	20
3.1.4	Analiza predykcyjna:.....	20
3.1.5	Zarządzanie danymi:	20
3.1.6	Teoria informacji:	20
3.1.7	Optymalizacja zasobów:	20
3.1.8	Etyka i prywatność danych:	20
3.1.9	Wizualizacja danych:.....	21
3.2	Narzędzia do analizy danych.....	21
3.2.1	Google Cloud IoT Core:	21
3.2.2	ThingSpeak:	21
3.2.3	AWS IoT Analytics:	21
4	Platformy Chmurowe	22
4.1	Protokół przesyłania danych Iot do platformy chmurowej	22
4.1.1	MQTT (Message Queuing Telemetry Transport):.....	22
4.1.2	HTTP/HTTPS:	22
4.1.3	CoAP (Constrained Application Protocol):	23
4.1.4	DDS (Data Distribution Service):	23
4.1.5	AMQP (Advanced Message Queuing Protocol):	23
4.1.6	Techniki bezprzewodowe:	23
4.2	Google Cloud Platform (GCP) IoT:	24
4.3	Amazon Web Services (AWS) IoT:.....	26
4.4	AWS IoT Analytics.....	27
4.4.1	Jak korzystać z AWS IoT Analytics	28
4.4.2	Pobieranie danych (<i>ang. Collect</i>)	29
4.4.3	Procesowanie (<i>ang. Process</i>).....	29



4.4.4	Przechowywanie danych (<i>ang. Store</i>)	30
4.5	Dostęp do interfejsu AWS IoT Analytics (ćwiczenia).....	32
4.5.1	Logowanie do konsoli AWS IoT Analytics	32
4.5.2	Tworzenie kanału z konsoli poleceń	34
4.5.3	Tworzenie magazynu danych	36
4.6	Microsoft Azure IoT Hub:	37
4.6.1	Uruchamianie Microsoft Azure (ćwiczenia)	37
4.6.2	Wysyłanie danych telemetrycznych z urządzenia do centrum IoT (ćwiczenia) 37	
4.7	IBM Watson IoT	42
4.7.1	Konfiguracja IBM Watson IoT (ćwiczenia)	47
4.8	Node.js.....	49
4.8.1	Przykład aplikacji Node.js.....	51
4.8.2	Instalacja Node.js	53
4.9	Wykorzystanie Node.js do implementacji warstwy WWW z czujnikami	55
4.9.1	Wybór odpowiedniego frameworka:	55
4.9.2	Komunikacja z czujnikami:.....	56
4.9.3	Przechowywanie danych:	56
4.9.4	Wizualizacja danych:.....	56
5	Przetwarzanie danych pojazdów w czasie rzeczywistym przy użyciu Internetu rzeczy (ćwiczenia)	58
5.1	Przepływ danych	59
5.2	Zastosowanie	61
6	Wnioski i podsumowanie	61
6.1	Amazon Web Services (AWS)	61
6.2	Microsoft Azure.....	62



6.3	Google Cloud Platform (GCP).....	62
6.4	IBM Cloud.....	62
6.5	Kryteria wyboru	62
7	Wykaz literatury	64
8	Wykaz tabel	64
9	Wykaz rysunków	68



1. Wstęp

1.1 Internet of Things (IoT),

czyli Internet Rzeczy, to koncepcja obejmująca połączenie urządzeń i systemów z internetem, umożliwiającą im zbieranie, przetwarzanie i wymianę danych. Termin ten odnosi się do szerokiego zakresu inteligentnych urządzeń, które są wbudowane w codzienne przedmioty i infrastrukturę, tworząc sieć połączonych obiektów. Obejmuje technologie czujników, RFID i technologie bezprzewodowe. Ponadto przenosi ogromną ilość informacji, które są pomocne w podejmowaniu inteligentnych decyzji i umożliwiają bezpieczeństwo, odporność, konwergencję, wydajność i przejrzystość w działaniu.

Urządzenia IoT obejmują różnorodne kategorie, takie jak inteligentne lodówki, samochody, systemy zarządzania budynkami, czujniki środowiskowe, urządzenia noszone (wearables) oraz wiele innych. Te inteligentne systemy wykorzystują zaawansowane technologie, takie jak czujniki, przetwarzanie danych w chmurze, sztuczna inteligencja (AI) i uczenie maszynowe, aby zbierać informacje o otoczeniu, analizować je i reagować w sposób dostosowany do potrzeb użytkowników.

W miarę jak technologia IoT staje się coraz bardziej rozpowszechniona, urządzenia te stają się nieodłącznym elementem naszego życia codziennego. Zapewniają one wygodę, efektywność energetyczną, bezpieczeństwo oraz możliwość optymalizacji różnych procesów i zadań. Jednakże, wraz z rosnącą liczbą połączonych urządzeń, pojawiają się również wyzwania związane z bezpieczeństwem danych, prywatnością oraz zarządzaniem złożonymi systemami.

1.2 Cloud Computing, czyli obliczeniowa chmura

to model dostarczania usług informatycznych, który charakteryzuje się następującymi cechami:

- **Dostępność przez Internet:** Usługi chmurowe są dostępne zdalnie za pośrednictwem internetu, co umożliwia korzystanie z nich z dowolnego miejsca i urządzenia.
- **Skalowalność:** Usługi chmurowe można łatwo skalować w górę lub w dół w zależności od aktualnych potrzeb, co pozwala na elastyczne dostosowywanie zasobów do zmieniających się wymagań.
- **Model płatności pay-as-you-go:** Koszty korzystania z usług chmurowych są zależne od rzeczywistego wykorzystania, co oznacza, że płaci się tylko za zużyte zasoby.
- Usługi chmurowe można podzielić na trzy główne kategorie:



- Infrastruktura jako usługa (*ang. infrastructure as a service IaaS*): Dostarcza klientom infrastrukturę obliczeniową, taką jak wirtualne maszyny, pamięć masową i sieci, bez konieczności inwestowania w własne centra danych.
- Platforma jako usługa (*ang. platform as a service PaaS*): Oferuje platformę programistyczną, która umożliwia tworzenie, testowanie i wdrażanie aplikacji bez konieczności zarządzania infrastrukturą.
- Oprogramowanie jako usługa (*ang. software as a service SaaS*): Dostarcza gotowe do użytku aplikacje, które są hostowane i zarządzane przez dostawcę usług chmurowych.

Rozwiązania Cloud Computing mogą być dostarczane w trzech modelach:

- Chmura prywatna: Infrastruktura budowana przez firmę na własne potrzeby, często w jej własnym centrum danych. Zapewnia pełną kontrolę nad danymi i zasobami.
- Chmura publiczna: Usługi dostępne dla wszystkich użytkowników internetu, oferowane przez dostawców takich jak Google Cloud Platform (GCP), Amazon Web Services (AWS) i Microsoft Azure.
- Chmura hybrydowa: Połączenie chmury prywatnej i publicznej, które umożliwia firmom korzystanie z zalet obu modeli, zachowując jednocześnie kontrolę nad krytycznymi danymi i zasobami.

Cloud Computing staje się coraz bardziej popularnym modelem dostarczania usług informatycznych, oferując firmom elastyczność, oszczędności kosztów i możliwość szybkiego skalowania. Jednak wiąże się również z wyzwaniami związanymi z bezpieczeństwem, zarządzaniem danymi i zgodnością z przepisami.

Chmura obliczeniowa (Cloud Computing) odgrywa kluczową rolę w umożliwianiu gromadzenia i przechowywania danych generowanych przez urządzenia Internetu Rzeczy (IoT). Oto, jak to działa:

- Zbieranie danych: Inteligentne urządzenia IoT, takie jak czujniki, kamery czy inteligentne liczniki, zbierają dane o otaczającym je środowisku.
- Przesyłanie danych: Zebrane dane są przesyłane do chmury obliczeniowej za pośrednictwem sieci internetowej. Może to odbywać się bezpośrednio z urządzeń IoT lub poprzez bramy (gateways) IoT.



- Przechowywanie danych: Chmura obliczeniowa zapewnia dużą pojemność przechowywania danych, co pozwala na gromadzenie ogromnych ilości informacji z wielu urządzeń IoT.
- Przetwarzanie danych: Chmura umożliwia przetwarzanie zebranych danych w czasie rzeczywistym lub w trybie wsadowym. Może to obejmować analizę danych, wykrywanie wzorców, uczenie maszynowe i inne zaawansowane operacje.
- Integracja z aplikacjami: Przetworzone dane mogą być udostępniane i wykorzystywane przez różne aplikacje biznesowe, systemy analityczne lub interfejsy użytkownika.
- Skalowalność: Chmura obliczeniowa zapewnia skalowalność, co pozwala na dostosowanie pojemności przechowywania i przetwarzania danych do aktualnych potrzeb.
- Bezpieczeństwo i zgodność: Chmura oferuje zaawansowane mechanizmy bezpieczeństwa, szyfrowania i zarządzania dostępem, co jest kluczowe dla ochrony wrażliwych danych IoT.

Dzięki chmurze obliczeniowej, urządzenia IoT mogą efektywnie zbierać i przetwarzać duże ilości danych, co umożliwia tworzenie inteligentnych systemów i usług opartych na analizie danych w czasie rzeczywistym, które są pomocne w podejmowaniu inteligentnych decyzji i umożliwiają bezpieczeństwo, odporność, konwergencję, wydajność i przejrzystość w działaniu.

Chmura czujników (Sensor Cloud) to innowacyjne podejście do zarządzania danymi czujników, które łączy możliwości chmury obliczeniowej z potrzebami sieci czujników (WSN - Wireless Sensor Networks). Oto główne cechy i zalety tej technologii:

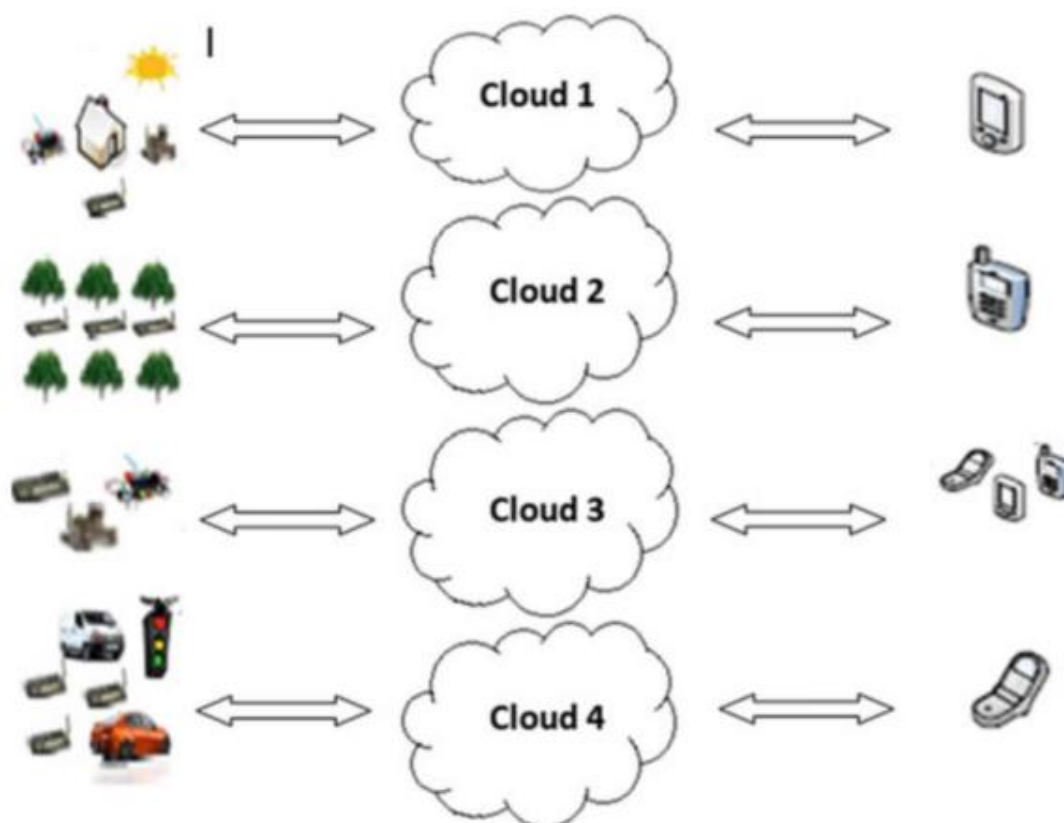
- Elastyczność i skalowalność: Chmura czujników umożliwia dynamiczne skalowanie zasobów w zależności od aktualnych potrzeb, co jest szczególnie ważne w przypadku zmiennych obciążeń danych czujników.
- Szybkość przetwarzania: Dzięki wykorzystaniu mocy obliczeniowej chmury, dane czujników mogą być przetwarzane w czasie rzeczywistym lub zbliżonym do rzeczywistego.
- Wizualizacja danych: Chmura czujników oferuje zaawansowane narzędzia do wizualizacji danych, co ułatwia analizę i interpretację informacji zbieranych przez czujniki.



- Programowalna analiza: Użytkownicy mogą tworzyć i uruchamiać własne algorytmy analizy danych czujników, co zwiększa elastyczność i możliwości dostosowania do konkretnych potrzeb.
- Zarządzanie danymi: Chmura czujników umożliwia efektywne zarządzanie dużymi ilościami danych czujników, w tym ich przechowywanie, indeksowanie i wyszukiwanie.
- Integracja z innymi usługami chmurowymi: Dane czujników mogą być łatwo zintegrowane z innymi usługami chmurowymi, takimi jak bazy danych, narzędzia analityczne czy systemy zarządzania danymi.
- Wydłużenie żywotności czujników: Dzięki efektywnemu przetwarzaniu i analizie danych, chmura czujników może optymalizować pracę czujników, co prowadzi do wydłużenia ich żywotności i zmniejszenia zużycia energii.
- Zastosowania w różnych dziedzinach: Chmura czujników znajduje zastosowanie w wielu obszarach, takich jak monitorowanie zdrowia, środowiska, infrastruktury, rolnictwa czy transportu.
- Zwiększona niezawodność: Dzięki redundancji i możliwości szybkiego skalowania, chmura czujników zapewnia większą niezawodność i dostępność danych.
- Koszty efektywności: Model płatności za rzeczywiste wykorzystanie (pay-as-you-go) pozwala na optymalizację kosztów, dostosowując je do aktualnych potrzeb.

Chmura czujników stanowi istotny krok w kierunku inteligentnego świata, łącząc możliwości sieci czujników z zaawansowanymi technologiami chmurowymi. Dzięki temu możliwe jest tworzenie bardziej zaawansowanych, elastycznych i efektywnych systemów monitorowania i analizy danych. Statystyki pokazują, że chmura czujników przewyższa tradycyjne sieci WSN i centra danych w chmurze, wydłużając żywotność czujników o 3,25% i zmniejszając zużycie energii o 36,68%. Są to cechy uważane za wymagane w inteligentnym świecie.

Na rysunku poniżej pokazano model chmury czujników



Rysunek 1 Model chmury czujników [4,7]

Dostawca sieci czujników (SNP - Sensor Network Provider) odgrywa kluczową rolę w architekturze chmury czujników. Jego zadania i funkcje można opisać następująco:

Zarządzanie siecią czujników: SNP zarządza całą infrastrukturą sieci czujników, w tym urządzeniami, oprogramowaniem i komunikacją między czujnikami.

Wszechobecna różnorodność czujników: SNP zapewnia szeroki zakres czujników, które mogą być zarówno statyczne (stacjonarne), jak i mobilne (ruchome). Czujniki te mogą zbierać różnorodne dane z otoczenia, takie jak temperatura, wilgotność, poziom zanieczyszczeń, ruch, itp.

Przesyłanie danych do chmury: SNP odpowiada za przesyłanie zebranych danych czujników do dostawcy usług w chmurze (CSP - Cloud Service Provider).

Zarządzanie zasilaniem: SNP musi zapewnić efektywne zarządzanie zasilaniem czujników, aby maksymalnie wydłużyć ich żywotność i zmniejszyć zużycie energii.

Bezpieczeństwo i prywatność danych: SNP odpowiada za zabezpieczenie danych czujników przed nieautoryzowanym dostępem i zapewnienie ich prywatności.

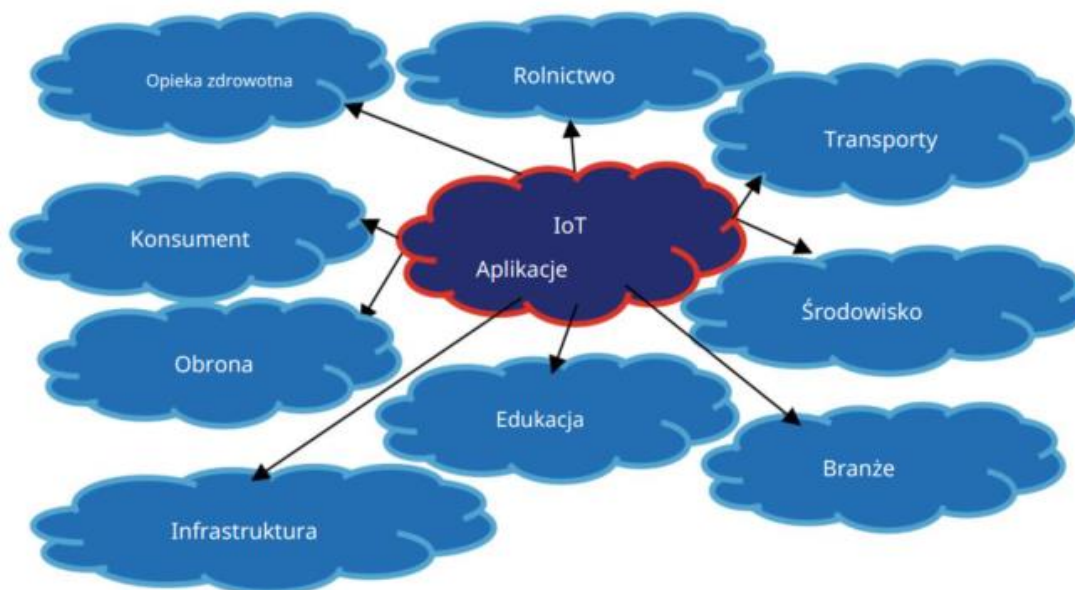
Integracja z usługami chmurowymi: SNP musi zapewnić kompatybilność sieci czujników z usługami chmurowymi oferowanymi przez CSP.

Dostawca usług w chmurze (CSP) natomiast odpowiada za:

- Przechowywanie danych czujników: CSP przechowuje przesłane dane czujników w swoich centrach danych, zapewniając ich bezpieczeństwo i dostępność.
- Przetwarzanie danych: CSP przetwarza dane czujników, wykorzystując moc obliczeniową chmury, aby zapewnić szybkie i efektywne przetwarzanie.
- Analiza danych: CSP może oferować zaawansowane narzędzia analityczne do analizy danych czujników, zapewniając użytkownikom końcowym (CSU) wartościowe informacje.
- Dostarczanie danych na żądanie: CSP udostępnia przetworzone dane czujników użytkownikom końcowym (CSU) na żądanie, zapewniając elastyczność i dostępność.
- Zarządzanie zasobami: CSP zarządza zasobami chmurowymi, takimi jak moc obliczeniowa, pamięć i przestrzeń dyskowa, aby zapewnić optymalne wykorzystanie i skalowalność.
- Współpraca między SNP, CSP i CSU jest kluczowa dla efektywnego funkcjonowania chmury czujników, zapewniając zbieranie, przetwarzanie i dostarczanie wartościowych danych w czasie rzeczywistym lub zbliżonym do rzeczywistego.

2 Internet rzeczy

IoT obejmuje zestaw urządzeń, maszyn lub sprzętu, takich jak urządzenia gospodarstwa domowego, pojazdy, sprzęt medyczny lub rolniczy, zawierający sprzęt (drukarki, monitory, serwery, siłowniki, czujniki itp.), oprogramowanie i łączącą sieć (tj. Internet) w celu skutecznej i zdalnej interakcji i komunikacji między tymi urządzeniami. Domena aplikacji IoT jest szeroko podzielona na następujące kategorie, jak pokazano na rysunku poniżej. [6,9].



Rysunek 2 Zastosowanie Internetu rzeczy [6,9]

Internet rzeczy (IoT) stanowi fundamentalną transformację w wielu branżach i rynkach. W ciągu ostatnich kilku lat nastąpił znaczący postęp w tej dziedzinie, który można opisać w następujący sposób:

2.1 Ewolucja od branżowych rozwiązań do globalnej wizji:

Początkowo IoT był rozwijany w konkretnych branżach, takich jak przemysł, medycyna czy transport. Obecnie następuje integracja i standaryzacja technologii IoT, co prowadzi do bardziej kompleksowej wizji łączności globalnego środowiska fizycznego.

2.2 Zwiększona integracja danych:

IoT generuje ogromne ilości danych z maszyn i czujników. Następuje integracja tych danych z istniejącymi źródłami danych, takimi jak systemy ERP, bazy danych rządowe i media społecznościowe. Pozwala to na uzyskiwanie nowych, praktycznych spostrzeżeń i wniosków.

2.3 Wzrost znaczenia analizy danych:

Analiza danych z IoT staje się kluczowym elementem w podejmowaniu decyzji biznesowych.

Umożliwia optymalizację procesów, prognozowanie konserwacji, personalizację usług i wiele innych zastosowań.



2.4 Rozwój technologii wspierających IoT:

Następuje postęp w dziedzinie sieci komórkowych (np. 5G), komunikacji bezprzewodowej (np. LoRaWAN, Sigfox) i technologii chmurowych. Rozwój tych technologii umożliwia bardziej niezawodną i efektywną komunikację między urządzeniami IoT.

2.5 Wzrost świadomości i akceptacji IoT:

Coraz więcej firm i organizacji dostrzega korzyści płynące z wdrożenia rozwiązań IoT.

Następuje wzrost akceptacji dla technologii IoT wśród konsumentów i decydentów biznesowych.

2.6 Wyzwania związane z bezpieczeństwem i prywatnością:

Wraz z rosnącą liczbą urządzeń IoT pojawiają się nowe zagrożenia związane z bezpieczeństwem i prywatnością danych.

Konieczne jest opracowanie i wdrożenie odpowiednich standardów i rozwiązań zabezpieczających.

2.7 Nowe modele biznesowe:

IoT prowadzi do powstania nowych modeli biznesowych, takich jak subskrypcje, usługi oparte na danych czy ekonomia współdzielenia.

Firmy muszą dostosować swoje modele biznesowe do nowych możliwości oferowanych przez IoT. Podsumowując, Internet rzeczy przekształca się z wąskich, branżowych rozwiązań w kompleksową wizję łączności globalnego środowiska fizycznego. Ta transformacja wymaga integracji danych, rozwoju technologii wspierających, zwiększonej świadomości i akceptacji, a także uwzględnienia wyzwań związanych z bezpieczeństwem i prywatnością. Jednocześnie IoT otwiera nowe możliwości biznesowe i operacyjne, które mogą znacząco poprawić efektywność i konkurencyjność firm w różnych branżach. Ważnym drogowskazem fundamentalnego znaczenia koncepcji IoT jest strategiczna działalność większości głównych dostawców ICT w zakresie opracowywania ofert IoT. Firmy, które znajdują się w centrum sektora telekomunikacyjnego, sieciowego, infrastruktury przemysłowej, systemów korporacyjnych i przetwarzania w chmurze, zbiegają się w strategii oferowania platform IoT w celu ułatwienia szerszej transformacji gospodarki w kierunku powszechnej łączności.



2.8 Datafication „Datafikacja” to nowa elektryfikacja

Chociaż IoT zasadniczo wiąże się z uniwersalną łącznością, obejmującą komputery osobiste i smartfony, większość obserwatorów omawiających IoT skupia się na możliwości połączenia istniejących maszyn, które wcześniej nie były połączone (na przykład silników samolotów), a także na radykalnym zwiększeniu liczby połączonych punktów w otoczeniu za pomocą czujników, siłowników i urządzeń, które nigdy nie zostałyby opracowane ani wdrożone bez podstawowej infrastruktury łączącej je w powszechną infrastrukturę ICT¹. Możemy postrzegać obecny nacisk na IoT jako analogiczny do transformacji infrastruktury elektrycznej ze specjalistycznych i odizolowanych systemów punktowych w powszechny, komodytyzowany² i niezbędny element nowoczesnego świata uprzemysłowionego. Od lat 80. XIX wieku do początku XX wieku nie było wszechogarniającej sieci elektrycznej, jaką mamy dzisiaj. Firmy i organizacje, które potrzebowały energii elektrycznej, wdrażały lokalne generatory, a „wiceprezesa ds. energii elektrycznej” byli potrzebni w organizacjach, aby nadzorować wysoce wyspecjalizowaną infrastrukturę. Jeśli energia elektryczna musiała zostać dostarczona do nowej części budynku lub obiektu, wiceprezes ds. energii elektrycznej wdrażał niestandardowy system. Obecnie elektryczność jest zasadniczo usługą towarową dostępną wszędzie w uprzemysłowionym świecie. Wszechobecna, znormalizowana sieć elektryczna dostarcza energię stosunkowo o wiele łatwiej. Ten typ abstrakcji dostęp do energii bez konieczności przeznaczania nadzwyczajnych zasobów na jej pozyskiwanie i zarządzanie nią — był kluczowym katalizatorem ogromnego wzrostu gospodarczego, jakiego świat doświadczył w latach 20. XX wieku. Firmy, organizacje i rządy mogą teraz skupić się na innowacjach, bez obciążeń i ograniczeń związanych z koniecznością skupienia się na podstawowych zasadach napędzających innowacje, które zostały opracowane wcześniej. Podobnie, poprzez szereg inicjatyw (w tym platformy chmurowe IoT, opracowywanie standardów, działania regulacyjne i tworzenie ekosystemu) następuje transformacja, która sprawi, że łączność stanie się powszechna, a co ważniejsze, dane z podłączonych maszyn i czujników staną się podstawową usługą, niemal jak towar. Dane są o wiele bardziej zróżnicowane i złożone niż elektryczność.

¹ ICT to skrót od Information and Communication Technologies, czyli technologie informacyjne i komunikacyjne. W skrócie, ICT to dziedzina obejmująca produkcję i wykorzystanie urządzeń telekomunikacyjnych i informatycznych, a także usługi z nimi związane.

² Komodytyzowany" to słowo, które oznacza utowarowienie, czyli przekształcenie czegoś z wartości użytkowej na wartość rynkową. W kontekście usług, komodytyzacja oznacza, że usługa staje się standaryzowana i wymienna, podobnie jak towary. Na przykład, komodytyzacja usług IT oznacza, że usługi te stają się tańsze i łatwiej dostępne, ale jednocześnie mniej zróżnicowane.



Uznaje się, że analogia „datafikacji” przedstawiona tutaj do transformacji „elektryfikacji” z XIX i początku XX wieku nie jest skorelowana pod każdym względem, ale analogia ta oddaje istotę idei odizolowanych, złożonych i ograniczonych systemów przekształcających się w wszechobecną użyteczność. Elektryfikacja miała fundamentalny wpływ na większość branż i rynków w XX wieku. Przewiduję się, że „datafikacja” będzie miała podobny wpływ w tym stuleciu.

Datafication to proces, w którym dane stają się kluczowym zasobem i źródłem wartości w organizacjach i społeczeństwie. Jest to zjawisko, które obejmuje kilka kluczowych aspektów:

- Generowanie danych: Wraz z rozwojem technologii, takich jak Internet Rzeczy (IoT), media społecznościowe, urządzenia mobilne i inne, generowane są ogromne ilości danych.
- Zbieranie i przechowywanie danych: Firmy i organizacje inwestują w infrastrukturę do zbierania, przechowywania i przetwarzania danych, taką jak centra danych, chmury obliczeniowe i systemy baz danych.
- Analiza danych: Rozwój zaawansowanych narzędzi analitycznych, takich jak sztuczna inteligencja, uczenie maszynowe i analiza predykcyjna, umożliwia wyciąganie wartościowych wniosków z dużych zbiorów danych.
- Wykorzystanie danych do podejmowania decyzji: Dane stają się podstawą do podejmowania świadomych decyzji biznesowych, operacyjnych i strategicznych.
- Tworzenie nowych produktów i usług: Dane są wykorzystywane do tworzenia nowych, spersonalizowanych produktów i usług, które lepiej odpowiadają potrzebom klientów.
- Zarządzanie danymi: Organizacje opracowują strategie zarządzania danymi, aby zapewnić ich jakość, bezpieczeństwo i prywatność.
- Współpraca i wymiana danych: Następuje integracja danych z różnych źródeł, w tym z otwartych baz danych rządowych, mediów społecznościowych i innych organizacji.
- Zmiana modeli biznesowych: Chociaż IoT zasadniczo wiąże się z uniwersalną łącznością, obejmującą komputery osobiste i smartfony, większość obserwatorów omawiających IoT skupia się na możliwości połączenia istniejących maszyn, które wcześniej nie były połączone (na przykład silników samolotów), a także na radykalnym zwiększeniu liczby połączonych punktów w otoczeniu za pomocą czujników,



siłowników i urządzeń, które nigdy nie zostałyby opracowane ani wdrożone bez podstawowej infrastruktury łączącej je w powszechną infrastrukturę ICT.

Datafication ma wpływ na różne aspekty życia społecznego, w tym edukację, opiekę zdrowotną, transport i zarządzanie miastami. Proces datafication jest napędzany przez rosnącą dostępność danych, postęp w technologiach analitycznych i chmurowych oraz rosnące zapotrzebowanie na wykorzystanie danych do tworzenia wartości biznesowej i społecznej. Jednocześnie wymaga on odpowiednich regulacji prawnych, standardów bezpieczeństwa i etycznych podejść do zarządzania danymi.

2.9 Wyzwania w realizacji powszechnego Internetu rzeczy

Na drodze do powszechnego stosowania Internetu Rzeczy (IoT) istnieje wiele wyzwań.

2.9.1 Zarządzanie złożonością

Rozdrobnione łańcuchy dostaw i ekosystemy utrudniają deweloperom IoT i klientom określenie optymalnych partnerów i źródeł narzędzi komponentów i usług pomocniczych. Na przykład producent OEM produktu wdrażający nową usługę produktu połączonego może musieć nawiązać nowe relacje z dostawcami komponentów częstotliwości radiowej (RF), licznymi operatorami komórkowymi w różnych krajach i różnymi dostawcami oprogramowania, aby umożliwić łączność i analizę danych.

Konieczność zmiany podstawowych procesów biznesowych lub organizacyjnych rodzi znaczną niepewność i ryzyko dla tradycyjnych organizacji wdrażających inicjatywy IoT. Jest to dramatyczna zmiana dla firmy zorganizowanej na bazie tradycyjnego producenta OEM, aby przyjąć procesy potrzebne do pomyślnego wdrożenia udanego, ciągłego modelu obsługi klientów.

Wielu tradycyjnych producentów OEM produktów nie ma doświadczenia w rozwijaniu połączonych produktów i usług. Kontynuując powyższy punkt, wiele tradycyjnych organizacji nie ma doświadczenia i wiedzy fachowej potrzebnej do pomyślnej integracji łączności ze swoimi produktami i usługami. To wyzwanie jest spotęgowane w przypadku organizacji niebędącej w branży telefonicznej — takiej jak przedsiębiorstwo użyteczności publicznej lub producent samochodów — aby przyjąć łączność bezprzewodową, zwłaszcza w przypadku tradycyjnie niepołączonego produktu.



Czasami niepewne środowiska regulacyjne zaostrzają potencjalną niepewność i ryzyko związane z angażowaniem się w projekty IoT. Na przykład dwie główne inicjatywy regulacyjne dotyczące pojazdów połączonych — inicjatywa śledzenia skradzionych pojazdów Contran 245 w Brazylii i inicjatywa UE dotycząca powiadamiania o wypadkach awaryjnych eCall — napotkały liczne opóźnienia we wdrażaniu. Okazało się to problematyczne dla firm próbujących zaplanować inicjatywy biznesowe i inwestycje związane z uchwaleniem tych przepisów.

2.9.2 Bezpieczeństwo i prywatność danych:

W miarę jak aplikacje IoT przenikają przemysł i społeczeństwo, aplikacje te tworzą coraz bardziej krytyczne zależności. Na przykład inteligentne sieci elektryczne, połączone samochody, inteligentne domy i wiele innych „inteligentnych” i „połączonych” aplikacji IoT naraża konsumentów i firmy na złośliwe ataki i wykorzystywanie. Urządzenia IoT generują ogromne ilości danych, które muszą być odpowiednio zabezpieczone. Konieczne jest opracowanie i wdrożenie skutecznych mechanizmów ochrony danych osobowych i wrażliwych informacji. Ciekawy przykład przypadku pochodzi z profilu Forbesa, w którym opisano, jak sprzedawca detaliczny był w stanie opracować techniki analityczne big data, które pozwoliły firmie ustalić, że nastolatka jest w ciąży na podstawie jej wzorców zakupowych i wysłać jej kupony na towary związane z ciążą, co wywołało duże zamieszanie u jej nieświadomej rodziny. Chociaż nie jest to dokładnie aplikacja IoT, pokazuje ona zarówno moc big data, jak i jednocześnie ich zdolność do ingerowania w życie osobiste. Bardziej konkretnie, z perspektywy IoT istnieją obawy, że hakerzy mogliby ukraść informacje, na przykład z inteligentnych liczników lub systemów automatyki domowej, aby określić, kiedy mieszkańcy domu są poza domem, co prowadzi do zwiększonego ryzyka włamania. Istnieje wiele innych przykładów, w jaki sposób prywatność jest potencjalnie zagrożona, gdy aplikacje IoT stają się coraz szerzej wdrażane.

2.9.3 Standardy i interoperacyjność:

Brak jednolitych standardów i protokołów komunikacyjnych utrudnia współpracę między różnymi urządzeniami i systemami IoT. Konieczne jest opracowanie uniwersalnych standardów, które umożliwią płynną komunikację między urządzeniami różnych producentów.



2.9.4 Infrastruktura sieciowa:

Wdrożenie na dużą skalę wymaga odpowiedniej infrastruktury sieciowej, w tym sieci komórkowych i bezprzewodowych. Konieczne są inwestycje w rozwój sieci, takich jak 5G, LoRaWAN³ czy Sigfox⁴, aby zapewnić niezawodną komunikację między urządzeniami. Sigfox to technologia komunikacji bezprzewodowej opracowana specjalnie dla zastosowań Internetu Rzeczy (IoT). Jest to rozwiązanie oparte na bardzo niskiej częstotliwości (Ultra Narrowband - UNB) i charakteryzuje się kilkoma kluczowymi cechami:

Bardzo niska częstotliwość (UNB): Sigfox wykorzystuje pasmo częstotliwości 868-868.2 MHz, co pozwala na bardzo niskie zużycie energii i niską moc transmisji.

Długi zasięg: Dzięki wykorzystaniu bardzo niskiej częstotliwości, Sigfox może osiągnąć zasięg do kilku kilometrów w terenie otwartym i nawet kilkunastu kilometrów w terenie zurbanizowanym.

Niskie zużycie energii: Urządzenia Sigfox charakteryzują się bardzo niskim zużyciem energii, co pozwala na długi czas pracy na bateriach (nawet do kilku lat).

2.9.5 Koszty wdrożenia:

Wdrożenie rozwiązań IoT może wiązać się z wysokimi kosztami początkowymi, w tym zakupem urządzeń, oprogramowania i infrastruktury. Konieczne jest opracowanie modeli biznesowych, które umożliwią firmom i organizacjom pokrycie tych kosztów.

2.9.6 Zarządzanie danymi:

Zbieranie i przetwarzanie dużych ilości danych wymaga odpowiednich narzędzi i kompetencji. Konieczne jest opracowanie strategii zarządzania danymi, w tym ich przechowywania, analizy i wykorzystania.

2.9.7 Regulacje prawne:

Wdrożenie IoT wymaga dostosowania do obowiązujących przepisów prawnych, w tym dotyczących ochrony danych osobowych, bezpieczeństwa i prywatności. Konieczne jest

³ LoRaWAN (Long Range Wide Area Network) to technologia komunikacji bezprzewodowej, która została opracowana specjalnie dla zastosowań Internetu Rzeczy (IoT). Jest to rozwinięcie oryginalnej technologii LoRa (Long Range), która została stworzona przez firmę Cognosoft.

⁴ Sigfox to firma technologiczna specjalizująca się w rozwiązaniach komunikacyjnych dla Internetu Rzeczy (IoT). Firma ta opracowała unikalną technologię komunikacji długiego zasięgu, która jest przeznaczona głównie dla urządzeń IoT o niskim zużyciu energii i niskiej przepustowości

opracowanie odpowiednich regulacji, które będą wspierać rozwój IoT, jednocześnie zapewniając ochronę interesów użytkowników.

2.9.8 Akceptacja społeczna:

Wdrożenie IoT na dużą skalę wymaga akceptacji społecznej i zaufania do nowych technologii. Konieczne są działania edukacyjne i informacyjne, aby przekonać społeczeństwo do korzyści płynących z IoT.

2.9.9 Złożoność systemów:

Systemy IoT są złożone i wymagają integracji wielu różnych komponentów. Konieczne jest opracowanie narzędzi i metodologii, które ułatwią projektowanie, wdrażanie i zarządzanie systemami IoT.

Pokonanie tych wyzwań wymaga współpracy między producentami urządzeń, dostawcami usług, organami regulacyjnymi, organizacjami branżowymi i społeczeństwem. Jednocześnie, rozwój technologii, takich jak sztuczna inteligencja, uczenie maszynowe i chmura obliczeniowa, może pomóc w przezwyciężeniu niektórych z tych przeszkód.

3 Analiza danych z IoT

Analiza danych z Internetu Rzeczy (IoT) jest złożonym i wieloaspektowym zagadnieniem, które wymaga interdyscyplinarnego podejścia, łączącego wiedzę z różnych dziedzin, takich jak informatyka, statystyka, uczenie maszynowe, teoria informacji i zarządzanie danymi. Poniżej przedstawiam naukowe aspekty analizy danych IoT:

3.1 Charakterystyka danych IoT:

Dane IoT charakteryzują się dużą objętością, różnorodnością i szybkością generowania. Często są to dane strumieniowe, wymagające przetwarzania w czasie rzeczywistym. Mogą zawierać niepełne, nieprecyzyjne lub zaszumione informacje.

3.1.1 Techniki przetwarzania danych:

Przetwarzanie danych IoT wymaga efektywnych algorytmów i struktur danych, takich jak drzewa skrócone, indeksy przestrzenne i algorytmy przybliżone. Konieczne jest opracowanie algorytmów odpornych na błędy i niepełne dane.



3.1.2 Analiza eksploracyjna:

Analiza eksploracyjna danych IoT obejmuje identyfikację wzorców, trendów i anomalii w dużych zbiorach danych. Wykorzystuje się techniki statystyczne, takie jak analiza skupień, analiza głównych składowych i analiza korelacji.

3.1.3 Uczenie maszynowe i sztuczna inteligencja:

Modele uczenia maszynowego, takie jak sieci neuronowe, lasy losowe i maszyny wektorów nośnych, są wykorzystywane do przewidywania przyszłych zdarzeń i trendów.

Konieczne jest opracowanie algorytmów odpornych na dane niepełne i zaszumione.

3.1.4 Analiza predykcyjna:

Analiza predykcyjna danych IoT obejmuje prognozowanie awarii, optymalizację procesów i personalizację usług. Wymaga to opracowania modeli predykcyjnych, które uwzględniają specyfikę danych IoT.

3.1.5 Zarządzanie danymi:

Zarządzanie danymi IoT obejmuje przechowywanie, archiwizację, udostępnianie i ochronę danych. Konieczne jest opracowanie efektywnych systemów zarządzania danymi, takich jak systemy zarządzania danymi (MDM) i bazy danych.

3.1.6 Teoria informacji:

Teoria informacji dostarcza narzędzi do oceny jakości danych, redukcji redundancji i kompresji danych. Wykorzystuje się koncepcje takie jak entropia, kodowanie źródłowe i kodowanie kanałowe.

3.1.7 Optymalizacja zasobów:

Analiza danych IoT wymaga efektywnego wykorzystania zasobów obliczeniowych i pamięciowych. Konieczne jest opracowanie algorytmów i struktur danych, które minimalizują zużycie zasobów.

3.1.8 Etyka i prywatność danych:

Analiza danych IoT wymaga uwzględnienia kwestii etycznych i prawnych związanych z prywatnością i bezpieczeństwem danych. Konieczne jest opracowanie mechanizmów szyfrowania, kontroli dostępu i zgodności z przepisami o ochronie danych osobowych.



3.1.9 Wizualizacja danych:

Wizualizacja danych IoT obejmuje tworzenie interaktywnych pulpitów nawigacyjnych i raportów, które ułatwiają interpretację i podejmowanie decyzji. Wykorzystuje się techniki wizualizacji danych, takie jak mapy ciepłe, wykresy czasowe i analiza przestrzenna.

Podsumowując, analiza danych z Internetu Rzeczy (IoT) jest złożonym zagadnieniem naukowym, które wymaga interdyscyplinarnego podejścia, łączącego wiedzę z informatyki, statystyki, uczenia maszynowego, teorii informacji i zarządzania danymi. Kluczowe jest opracowanie odpowiednich technik przetwarzania, analizy i wizualizacji danych, które uwzględniają specyfikę danych IoT i umożliwiają uzyskanie wartościowych wniosków dla organizacji.

3.2 Narzędzia do analizy danych

3.2.1 Google Cloud IoT Core:

Jest to usługa w chmurze Google, która umożliwia bezpieczne podłączanie urządzeń IoT do chmury i przetwarzanie wiadomości. Charakteryzuje się łatwością rejestracji urządzeń i szybkim wdrażaniem. Umożliwia zbieranie wielu strumieni danych, co ułatwia zarządzanie danymi. Integruje się z innymi usługami Google Cloud, takimi jak BigQuery do analizy danych.

3.2.2 ThingSpeak:

Jest to rozwiązanie open-source, które umożliwia szybkie budowanie prototypów IoT. Oferuje możliwość analizowania i wizualizacji danych za pomocą widgetów MATLAB. Jest przyjazne dla początkujących i oferuje łatwy w użyciu interfejs. Może być ograniczone w skalowalności i funkcjonalnościach w porównaniu z bardziej zaawansowanymi platformami.

3.2.3 AWS IoT Analytics:

Jest to usługa w chmurze Amazon Web Services (AWS), która umożliwia wykonywanie złożonych analiz na dużych ilościach danych IoT. Oferuje zaawansowane możliwości przetwarzania danych, w tym transformację danych, agregację i wykrywanie wzorców. Integruje się z innymi usługami AWS, takimi jak Amazon S3, Amazon Redshift i Amazon Athena. Charakteryzuje się dużą skalowalnością i elastycznością, ale może być bardziej złożona w konfiguracji dla początkujących.



Oprócz wymienionych narzędzi, istnieją również inne popularne platformy do analizy danych IoT, takie jak:

- Microsoft Azure IoT Hub: Usługa w chmurze Microsoftu, która umożliwia zbieranie, przetwarzanie i analizę danych IoT.
- IBM Watson IoT Platform: Platforma IBM, która oferuje zaawansowane możliwości analizy danych i wizualizacji.
- Splunk: Narzędzie do analizy danych i monitorowania, które może być wykorzystane do analizy danych IoT.

Wybór odpowiedniego narzędzia do analizy danych IoT zależy od konkretnych wymagań projektu, takich jak skalowalność, funkcjonalność, łatwość użycia, integracja z istniejącymi systemami oraz koszty. Warto również wziąć pod uwagę preferencje dotyczące ekosystemu chmurowego i doświadczenie zespołu w pracy z danym narzędziem.

4 Platformy Chmurowe

4.1 Protokół przesyłania danych Iot do platformy chmurowej

Protokoły przesyłania danych z urządzeń IoT do platformy chmurowej mogą się różnić w zależności od konkretnej platformy i wymagań aplikacji. Oto kilka popularnych protokołów i technik używanych do przesyłania danych IoT do platformy chmurowej:

4.1.1 MQTT (Message Queuing Telemetry Transport):

Jest to lekki, efektywny protokół komunikacyjny, który jest często używany w aplikacjach IoT. Obsługuje publikowanie/subskrybowanie wiadomości, co ułatwia komunikację między urządzeniami. MQTT jest często wykorzystywany do przesyłania danych z urządzeń IoT do platform chmurowych, takich jak AWS IoT Core, Azure IoT Hub i IBM Watson IoT Platform.

4.1.2 HTTP/HTTPS:

Protokół HTTP (Hypertext Transfer Protocol) lub jego bezpieczna wersja HTTPS (Hypertext Transfer Protocol Secure) są często używane do przesyłania danych z urządzeń IoT do platform chmurowych. Można wykorzystać RESTful API udostępniane przez platformę chmurową do wysyłania danych.



4.1.3 CoAP (Constrained Application Protocol):

Jest to protokół komunikacyjny zaprojektowany specjalnie dla urządzeń o ograniczonych zasobach, takich jak urządzenia IoT. CoAP jest lekki i efektywny, co czyni go odpowiednim do przesyłania danych z urządzeń IoT do platform chmurowych.

4.1.4 DDS (Data Distribution Service):

Jest to protokół komunikacyjny stworzony przez OMA (Open Mobile Alliance) do dystrybucji danych w czasie rzeczywistym. DDS jest często używany w aplikacjach IoT wymagających wysokiej wydajności i niezawodności.

4.1.5 AMQP (Advanced Message Queuing Protocol):

Jest to zaawansowany protokół komunikacyjny oparty na modelu publikowania/subskrybowania. AMQP jest często używany w aplikacjach IoT, szczególnie w przypadku wymagających dużej przepustowości i niezawodności.

4.1.6 Techniki bezprzewodowe:

Wiele urządzeń IoT wykorzystuje techniki bezprzewodowe do przesyłania danych, takie jak Wi-Fi, Bluetooth, LoRaWAN, Sigfox i inne. Dane z urządzeń bezprzewodowych są następnie przekazywane do platformy chmurowej za pośrednictwem bramki lub innego pośrednika.

Wybór odpowiedniego protokołu lub techniki przesyłania danych zależy od wielu czynników, takich jak:

- Wymagania dotyczące wydajności i opóźnień
- Ograniczenia zasobów urządzeń IoT
- Wymagania dotyczące bezpieczeństwa i prywatności
- Integracja z istniejącymi systemami i protokołami
- Wsparcie ze strony platformy chmurowej

Wiele platform chmurowych, takich jak AWS IoT Core, Azure IoT Hub i IBM Watson IoT Platform, oferuje obsługę różnych protokołów i technik przesyłania danych, co ułatwia integrację z różnymi urządzeniami i systemami IoT. Warto również wziąć pod uwagę specyficzne wymagania aplikacji i preferencje dotyczące ekosystemu chmurowego przy wyborze odpowiedniego protokołu lub techniki przesyłania danych.

W tabeli poniżej przedstawiono protokoły stosowane przez platformy chmurowe



Tabela 1 Protokoły stosowane przez platformy chmurowe

L.P.	Platforma	Protokoły
1	Microsoft Azure	MQTT, AMQP, HTTPS, MQTT poprzez websockets, AMQP poprzez websockets
2	AWS	MQTT, HTTPS, Websockets, LoRaWAN
3	IBM Bluemix	MQTT, HTTP, HTTPS
4	Thingworx	MQTT, HTTPS, HTTP, COAP
5	Google	MQTT, HTTP

Wybór odpowiedniej platformy chmurowej do przechowywania i przetwarzania danych z IoT może być wyzwaniem ze względu na różnorodność dostępnych rozwiązań. Na podstawie przedstawionych informacji, oto krótkie porównanie trzech głównych platform:

4.2 Google Cloud Platform (GCP) IoT:

Zalety:

- Kompletny zestaw narzędzi do przetwarzania danych IoT
- Skalowalna integracja w czasie rzeczywistym
- Możliwość szybkiego wglądu w wartościowe wnioski i monitorowanie trendów rynkowych
- Wsparcie dla uczenia maszynowego (Vertex AI)
- Narzędzia do wizualizacji danych (Google Data Studio)

Wady:

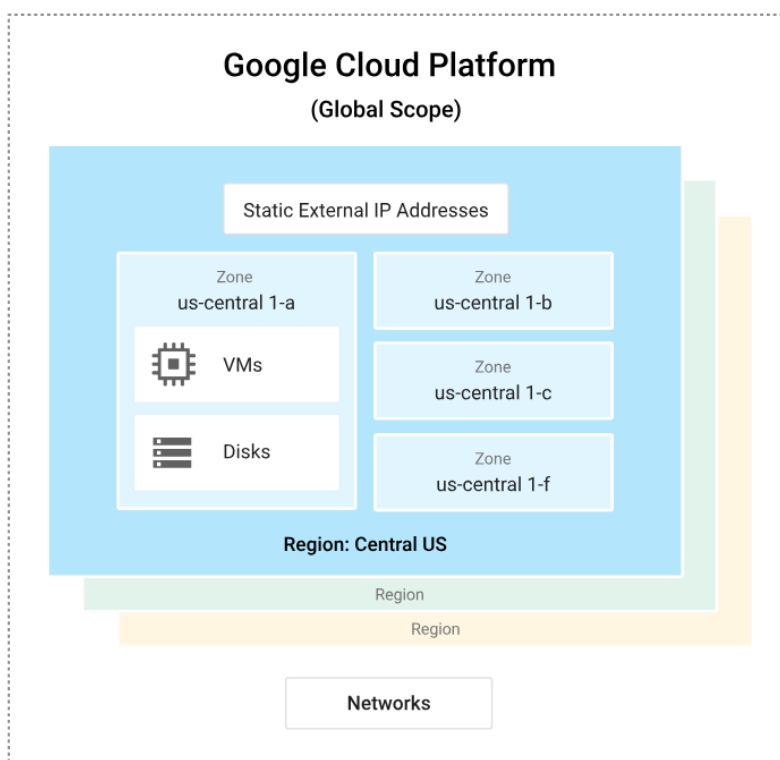
- Może być mniej znana wśród niektórych użytkowników. Program został zakończony i wycofany przez Google.⁵

Obecnie oferowana jest usługa Google Cloud, składa się z zestawu zasobów fizycznych, takich jak komputery i dyski twarde, oraz zasobów wirtualnych, takich jak maszyny wirtualne (VM), zlokalizowanych w centrach danych na całym świecie. Każda lokalizacja centrum danych znajduje się w regionie. Regiony są dostępne w Azji, Australii, Europie, Afryce, na Bliskim Wschodzie, w Ameryce Północnej i Południowej. Każdy region to zbiór stref, które są izolowane od siebie w ramach regionu. Każda strefa jest identyfikowana przez nazwę, która łączy identyfikator literowy z nazwą regionu. Na przykład strefa w regionie Azja Wschodnia

⁵ <https://cloud.google.com/iot-core>



nosi nazwę asia-east1-a. Ten rozkład zasobów zapewnia szereg korzyści, w tym redundancję w przypadku awarii i zmniejszone opóźnienia poprzez umieszczanie zasobów bliżej klientów. Ten rozkład wprowadza również pewne zasady dotyczące sposobu, w jaki zasoby mogą być używane razem. W chmurze obliczeniowej to, co możesz uważać za oprogramowanie i sprzęt, staje się usługami. Usługi te zapewniają dostęp do podstawowych zasobów. Lista dostępnych usług Google Cloud⁶ jest ciągle uaktualniana. Na rysunku poniżej pokazano strukturę Google Cloud.



Rysunek 3 Globalna struktura Google Cloud źródło <https://cloud.google.com>

Na rysunku poniżej pokazano przykładową architekturę platformy IoT działającą w chmurze. Widzimy jak platforma IoT wdraża brokera MQTT lub punkt końcowy do łączności urządzeń. Platforma IoT jest połączona z zewnętrznym proxy Network Load Balancer w celu dystrybucji ruchu z urządzeń brzegowych. Dodatkowe aplikacje IoT mogą łączyć się z platformą IoT za pośrednictwem Pub/Sub lub za pomocą łącznika Dataflow MQTT. Platforma IoT zapewnia zestaw usług zarządzania urządzeniami. Jak pokazano na schemacie, usługi te są następujące:

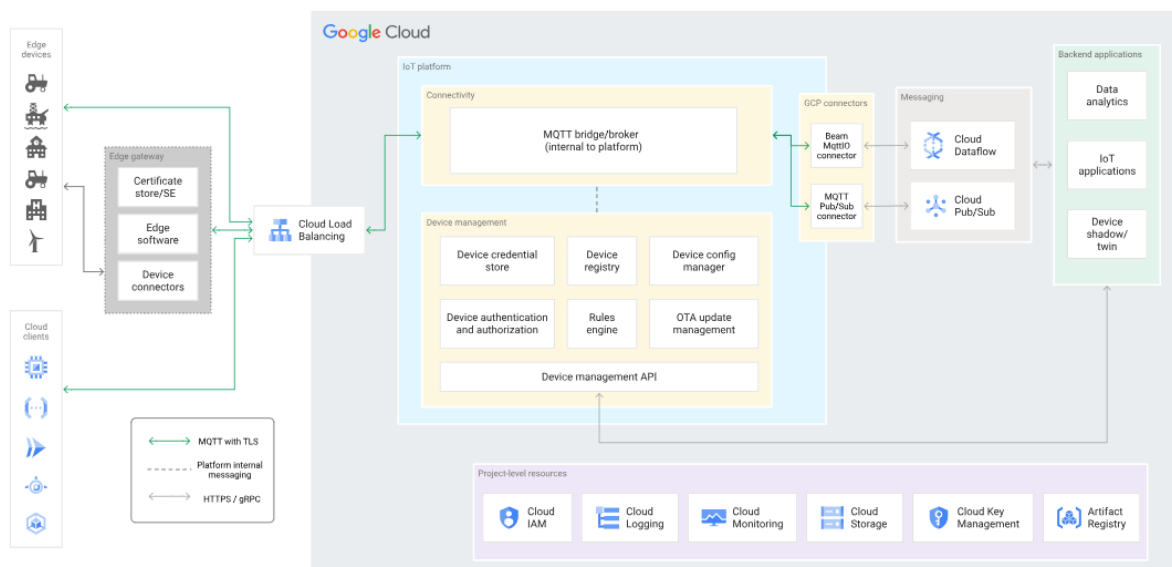
⁶ <https://cloud.google.com/products>



Magazyn danych uwierzytelniających urządzenia

Silnik reguł

- Uwierzytelnianie i autoryzacja urządzenia
- Zarządzanie konfiguracją urządzeń
- Rejestr urządzeń
- Zarządzanie aktualizacjami urządzeń



Rysunek 4 Architektura z ogólnym produktem platformy IoT działającym w Google Cloud
źródło <https://cloud.google.com/architecture/connected-devices/iot-platform-product-architecture>

4.3 Amazon Web Services (AWS) IoT:

Zalety:

- Bardzo skalowalna platforma (potencjalnie miliardy urządzeń)
- Wszechstronny zestaw narzędzi do przetwarzania danych IoT
- Bezpieczne przesyłanie wiadomości
- Integracja z usługami ML i analizy danych
- Zestaw SDK do łatwiejszej integracji i tworzenia aplikacji

Wady:

- Może być droższa w porównaniu do niektórych konkurencyjnych rozwiązań

4.4 AWS IoT Analytics

AWS IoT Analytics to usługa w chmurze Amazon Web Services (AWS), która umożliwia przetwarzanie i analizę dużych ilości danych z Internetu Rzeczy (IoT). AWS IoT Analytics to usługa, która automatyzuje proces analizy danych z urządzeń IoT. Dzięki niej możesz filtrować i przekształcać dane, wybierając tylko potrzebne informacje z urządzeń i stosując transformacje matematyczne, aby je przetworzyć. Możesz również wzbogacić dane o metadane specyficzne dla urządzenia, np. typ czy lokalizację, aby uzyskać pełniejszy obraz. Przetworzone dane możesz zapisać w magazynie danych szeregów czasowych, co ułatwi ich analizę. Następnie możesz analizować dane, wykorzystując wbudowany silnik zapytań SQL do analizy danych lub przeprowadzając bardziej zaawansowane analizy uczenia maszynowego. AWS IoT Analytics umożliwia także eksplorację danych poprzez integrację z Jupyter Notebook⁷ oraz wizualizację danych poprzez integrację z Amazon QuickSight⁸. Dane IoT, pochodzące z urządzeń rejestrujących informacje takie jak temperatura, ruch czy dźwięk, często charakteryzują się brakami, uszkodzeniami i fałszywymi odczytami. Aby przeprowadzić analizę, dane te wymagają wstępnego oczyszczenia. Dodatkowo, ich znaczenie często zależy od kontekstu, który dostarczają dane zewnętrzne. AWS IoT Analytics rozwiązuje te problemy, umożliwiając zbieranie, przetwarzanie i przechowywanie dużych ilości danych z urządzeń. Narzędzie to pozwala na przeszukiwanie i analizę zebranych danych, a także zawiera gotowe modele dla typowych przypadków użycia IoT. Dzięki temu można odpowiadać na pytania dotyczące potencjalnych awarii urządzeń czy ryzyka porzucenia urządzeń noszonych przez klientów. Jest to zaawansowane narzędzie do analizy danych IoT, które oferuje szereg funkcji i możliwości:

Przetwarzanie danych:

AWS IoT Analytics umożliwia przetwarzanie danych IoT w czasie rzeczywistym lub wsadowo. Oferuje możliwość transformacji danych, agregacji, filtrowania i ekstrakcji informacji.

- **Analiza danych:** Usługa umożliwia wykonywanie złożonych analiz na danych IoT, takich jak wykrywanie wzorców, prognozowanie i analiza predykcyjna. Oferuje wbudowane funkcje do analizy danych, takie jak sumowanie, średnia, maksimum, minimum i inne.

⁷ <http://jupyter.org>

⁸ <https://quicksight.aws>



- Wyzwalanie akcji: AWS IoT Analytics może wyzwać akcje na podstawie wyników analizy danych, takie jak wysyłanie powiadomień, uruchamianie skryptów lub wywoływanie innych usług AWS.
- Integracja z innymi usługami AWS: Usługa łatwo integruje się z innymi usługami AWS, takimi jak Amazon S3 do przechowywania danych, Amazon Redshift do analizy danych, Amazon Kinesis do przetwarzania strumieniowego i Amazon CloudWatch⁹ do monitorowania.
- Skalowalność: AWS IoT Analytics jest skalowalny i może obsługiwać duże ilości danych IoT. Można skalować zasoby w górę lub w dół w zależności od aktualnych potrzeb.
- Bezpieczeństwo i prywatność: Usługa oferuje mechanizmy zabezpieczania danych, takie jak szyfrowanie danych w spoczynku i w transmisji oraz kontrola dostępu. Zapewnia zgodność z przepisami o ochronie danych osobowych, takimi jak GDPR.
- Wizualizacja danych: AWS IoT Analytics może być zintegrowany z narzędziami do wizualizacji danych, takimi jak Amazon QuickSight, aby tworzyć interaktywne pulpity nawigacyjne i raporty.
- Łatwość użycia: Usługa oferuje przyjazny interfejs użytkownika, który umożliwia konfigurację przepływów pracy i analiz danych bez konieczności pisania kodu.

Dostępne są również API i SDK, które umożliwiają integrację z własnymi aplikacjami.

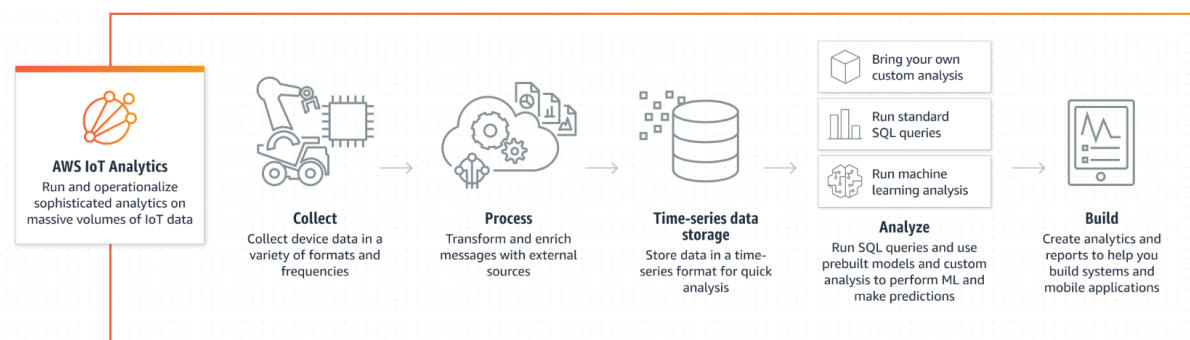
AWS IoT Analytics jest przydatnym narzędziem do analizy danych IoT, które oferuje zaawansowane możliwości przetwarzania i analizy danych w chmurze. Jest szczególnie przydatne dla organizacji, które potrzebują wykonywać złożone analizy na dużych ilościach danych IoT i chcą korzystać z ekosystemu AWS. Jednak ze względu na swoją zaawansowaną funkcjonalność, może być bardziej złożony w konfiguracji i wymagać większego doświadczenia technicznego niż niektóre inne narzędzia do analizy danych IoT.

4.4.1 Jak korzystać z AWS IoT Analytics

Na rysunku poniżej pokazano wykorzystanie usługi AWS IoT Analytics¹⁰

⁹ <https://aws.amazon.com/cloudwatch/>

¹⁰ <https://aws.amazon.com/iot-analytics/>



Rysunek 5 wykorzystanie usługi AWS IoT Analytics źródło: <https://aws.amazon.com/iot-analytics>

4.4.2 Pobieranie danych (ang. *Collect*)

Integracja z AWS IoT Core¹¹ — usługa AWS IoT Analytics jest w pełni zintegrowana z AWS IoT Core, dzięki czemu może odbierać wiadomości z podłączonych urządzeń w trakcie ich przesyłania strumieniowego.

Użycie interfejsu API wsadowego, aby dodać dane z dowolnego źródła — usługa AWS IoT Analytics może odbierać dane z dowolnego źródła przez HTTP. Oznacza to, że każde urządzenie lub usługa podłączona do Internetu może wysyłać dane do AWS IoT Analytics.¹²

•Zbieranie tylko tych danych, które chcesz przechowywać i analizować — można użyć narzędzia konsola AWS IoT Analytics do konfiguracji AWS IoT Analytics w celu odbierania wiadomości z urządzeń za pośrednictwem filtrów tematów MQTT w różnych formatach i częstotliwościach. AWS IoT Analytics weryfikuje, czy dane mieszczą się w określonych parametrach zdefiniowanych przez użytkownika i tworzy kanały. Następnie usługa jest kierowana do odpowiednich potoków w celu przetwarzania, transformacji i wzbogacania wiadomości

4.4.3 Procesowanie (ang. *Process*)

Oczyszczanie i filtrowanie — usługa AWS IoT Analytics umożliwia definiowanie funkcji AWS Lambda, które są

Przechowywanie przetworzonych i surowych danych — usługa AWS IoT Analytics przechowuje przetworzone dane, a także wyzwalane, gdy AWS IoT Analytics wykrywa

¹¹ <https://aws.amazon.com/iot-core/>

¹² https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_BatchPutMessage.html

brakujące dane, dzięki czemu możesz uruchomić kod, aby oszacować i wypełnić luki. Można również zdefiniować maksymalne i minimalne filtry oraz progi percentylowe¹³, aby usunąć wartości odstające w swoich danych.

Transformacja — usługa AWS IoT Analytics przekształca wiadomości, korzystając ze zdefiniowanej przez Ciebie logiki matematycznej lub warunkowej, dzięki czemu możesz wykonywać typowe obliczenia, takie jak konwersja stopni Celsjusza na stopnie Fahrenheita.

Wzbogacanie — usługa AWS IoT Analytics może wzbogacać dane o zewnętrzne źródła danych, takie jak np. prognoza pogody, a następnie przekierowuje dane do magazynu danych AWS IoT Analytics.

4.4.4 Przechowywanie danych (*ang. Store*)

Magazyn danych szeregów czasowych — AWS IoT Analytics przechowuje dane urządzenia w zoptymalizowanym czasie w pamięci, dla szybszego pobierania i analizy. Można również zarządzać uprawnieniami dostępu, wdrażać zasady przechowywania danych i eksportować swoje dane do zewnętrznych punktów dostępu.

Przechowywanie przetworzonych i surowych danych — usługa AWS IoT Analytics przechowuje przetworzone dane, a także automatycznie przechowuje surowe dane pobrane, dzięki czemu można je przetworzyć w późniejszym czasie.

może odbierać wiadomości z podłączonych urządzeń w trakcie ich przesyłania strumieniowego.

Analizowanie danych (*ang. analyze*)

Uruchomienie zapytania SQL Ad-hoc¹⁴ — AWS IoT Analytics udostępnia silnik zapytań SQL, dzięki czemu można uruchamiać zapytania ad-hoc i szybko uzyskiwać wyniki. Usługa umożliwia korzystanie ze standardowych zapytań SQL w celu wyodrębnienia danych z magazynu danych, aby odpowiedzieć na pytania takie jak np. średnia odległość przebyta przez flotę połączonych pojazdów lub ile drzwi w inteligentnym budynku jest zamkniętych po

¹³ Percentylowy rozkład danych to sposób przedstawienia rozkładu danych, który pokazuje, jaki procent danych znajduje się poniżej określonej wartości. Jest to użyteczne narzędzie do analizy i interpretacji danych, szczególnie w przypadku danych o rozkładzie niesymetrycznym lub z wieloma wartościami odstającymi.

¹⁴ SQL Ad-hoc (doraźne zapytania SQL) to termin odnoszący się do tworzenia i wykonywania zapytań SQL w sposób doraźny, bez wcześniejszego planowania lub przygotowania. Jest to przeciwieństwo zapytań przygotowanych (prepared statements) i zapytań przechowywanych (stored procedures).



godzinie 15.00. Zapytania te można ponownie wykorzystać, nawet jeśli podłączone urządzenia, wielkość floty i wymagania analityczne ulegną zmianie.

Analiza szeregów czasowych — usługa AWS IoT Analytics obsługuje analizę szeregów czasowych, dzięki czemu możesz analizować wydajność urządzeń na przestrzeni czasu i zrozumienie, jak i gdzie są one wykorzystywane, ciągłe monitorowanie danych urządzeń w celu przewidywania problemów konserwacyjnych oraz monitorowanie czujników w celu przewidywania warunków środowiskowych i reagowania na nie.

Hostowane notatniki do zaawansowanej analityki i uczenia maszynowego — AWS IoT Analytics obejmuje obsługę hostowanych notatników w Jupyter Notebook¹⁵ do analizy statystycznej i uczenia maszynowego. Usługa obejmuje zestaw szablonów notatników, które zawierają modele uczenia maszynowego i wizualizacje autorstwa AWS. Można użyć szablonów, aby rozpocząć pracę z przypadkami użycia IoT związanymi z profilowaniem awarii urządzeń, prognozowaniem zdarzeń, takich jak niskie wykorzystanie, które mogą sygnalizować, że klient porzuci produkt, lub segmentacją urządzeń według poziomów wykorzystania przez klienta (na przykład intensywnych użytkowników, użytkowników weekendowych) lub kondycji urządzenia.

Po utworzeniu notatnika możesz umieścić go w kontenerze i wykonywać zgodnie z określonym przez siebie harmonogramem.

Prognozowanie — klasyfikację statystyczną można przeprowadzić za pomocą metody zwanej regresją logistyczną. Można również użyć Long-Short-Term Memory (LSTM), która jest potężną techniką sieci neuronowych do przewidywania wyników lub stanu procesu, który zmienia się w czasie. Wstępnie zbudowane szablony notatników obsługują również algorytm klastrowania K-means¹⁶ do segmentacji urządzeń, który grupuje urządzenia w kohorty podobnych urządzeń. Te szablony są zazwyczaj używane do profilowania kondycji i stanu urządzeń, takich jak jednostki HVAC¹⁷ w fabryce czekolady lub zużycie łopatek turbiny wiatrowej. Ponownie, te szablony notatników mogą być zawarte i wykonywane zgodnie z harmonogramem.

¹⁵ <https://jupyter.org>

¹⁶ Algorytm K-średnich (z ang. k-means) inaczej zwany również algorytmem centroidów, służy do podziału danych wejściowych na z góry założoną liczbę k skupień. Algorytm ten polega na przenoszeniu obiektów ze skupienia do skupienia, aż do momentu, gdy wszystkie obiekty w każdym skupieniu będą bliżej swojego centroidu niż centroidów innych skupień.

¹⁷ Jednostki HVAC to skrót od Heating, Ventilation and Air Conditioning, czyli ogrzewanie, wentylacja i klimatyzacja

Tworzenie i wizualizowanie (ang. Build and visualize)

Integracja z Amazon QuickSight¹⁸ — AWS IoT Analytics udostępnia łącznik z Amazon QuickSight, dzięki czemu można wizualizować zestawy danych na pulpicie QuickSight.

Integracja z konsolą — można również wizualizować wyniki lub analizy ad-hoc w osadzonym notatniku Jupyter Notebook w konsoli AWS IoT Analytics.

4.5 Dostęp do interfejsu AWS IoT Analytics (ćwiczenia)

Jako część AWS IoT usługa AWS IoT Analytics udostępnia interfejsy umożliwiające urządzeniom generowanie danych i aplikacjom interakcję z generowanymi danymi: Uruchom polecenia dla AWS IoT Analytics w systemach Windows, OS X i Linux. Te polecenia umożliwiają tworzenie i zarządzanie rzeczami, certyfikatami, regułami i zasadami:

Postępuj zgodnie z poniższymi instrukcjami, aby utworzyć kanał, potok, magazyn danych i zestaw danych AWS IoT

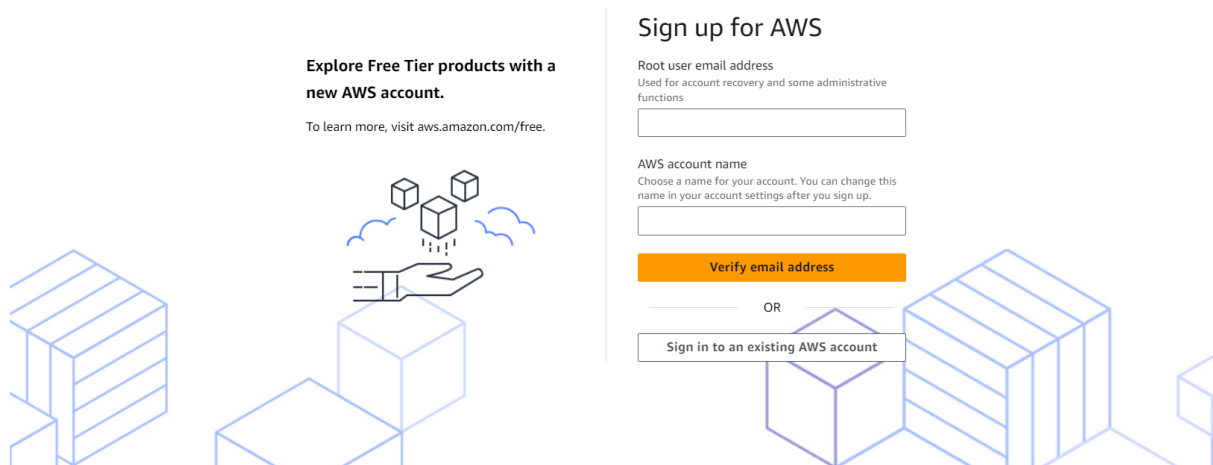
- Zaloguj się do konsoli AWS IoT Analytics
- Utwórz magazyn danych
- Utwórz potok
- Wysyłaj dane wiadomości za pomocą AWS IoT
- Sprawdź postęp wiadomości AWS IoT
- Dostęp do wyników zapytania
- Przeglądaj swoje dane
- Szablony notatników

4.5.1 Logowanie do konsoli AWS IoT Analytics

Aby utworzyć konto AWS:

- Otwórz <https://portal.aws.amazon.com/billing/signup> rysunek poniżej

¹⁸ <https://aws.amazon.com/quicksight>



Rysunek 6 Strona logowania AWS

- Postępuj zgodnie ze wskazówkami
- Zaloguj się do konsoli zarządzania AWS i przejdź do strony <https://console.aws.amazon.com/iotanalytics/>

Aby utworzyć kanał:

- W <https://console.aws.amazon.com/iotanalytics/>, w sekcji Przygotuj dane za pomocą AWS IoT Analytics wybierz opcję *Wyświetl kanały* rysunek poniżej.



Rysunek 7 Przygotowanie kanału

- Na stronie Kanały wybierz opcję Utwórz kanał.
- Na stronie *Określ szczegóły kanału* wprowadź szczegóły dotyczące swojego kanału.



- Wprowadź nazwę kanału, która jest unikalna i którą możesz łatwo zidentyfikować.
- (Opcjonalnie) W przypadku tagów dodaj jeden lub więcej niestandardowych tagów (pary klucz-wartość) do swojego kanału. Tagi mogą pomóc Ci zidentyfikować zasoby, które tworzysz dla AWS IoT Analytics.
- Wybierz Dalej.

AWS IoT Analytics przechowuje Twoje surowe, nieprzetworzone dane urządzeń IoT w kontenerze Amazon Simple Storage Service (Amazon S3). Możesz wybrać własny kontener Amazon S3, do którego możesz uzyskać dostęp i zarządzać nim, lub AWS IoT Analytics może zarządzać kontenerem Amazon S3 za Ciebie.

- w polu Typ magazynu wybierz Magazyn zarządzany przez usługę.
- W polu Wybierz okres przechowywania danych surowych wybierz Bezterminowo.
- Wybierz Dalej.
- Na stronie Konfiguruj źródło wprowadź informacje dotyczące usługi AWS IoT Analytics w celu zbierania wiadomości dane z AWS IoT Core.
- Wprowadź filtr tematu AWS IoT Core, na przykład update/environment/dht1. Później w tym samouczku użyjesz tego filtra tematu, aby wysłać dane wiadomości do swojego kanału.
- W obszarze roli IAM wybierz Utwórz nową. W oknie Utwórz nową rolę wprowadź nazwę
- Wybierz Dalej.
- Przejrzyj swoje wybory i wybierz opcję Utwórz kanał.
- Sprawdź, czy Twój nowy kanał jest widoczny na stronie Kanały .

4.5.2 Tworzenie kanału z konsoli poleceń¹⁹

Kanał zbiera i archiwizuje surowe, nieprzetworzone dane wiadomości przed opublikowaniem ich w kanale. Wiadomości przychodzące są wysyłane do kanału, więc pierwszym krokiem jest utworzenie kanału dla danych.

Wykonujemy polecenie

¹⁹<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html>



```
aws iotanalytics create-channel --channel-name mychannel
```

Rysunek 8 polecenie tworzenia kanału

Aby wyświetlić listę kanałów

```
aws iotanalytics describe-channel --channel-name mychannel
```

Aby uzyskać więcej informacji o kanale.

```
aws iotanalytics describe-channel --channel-name mychannel
```

Nieprzetworzone wiadomości kanałowe są przechowywane w kontenerze Amazon S3 zarządzanym przez AWS IoT Analytics, lub w zarządzanym przez Ciebie. Użyj parametru **channelStorage**, aby określić, który kontener wybierasz. Domyślnie jest to zarządzany przez usługę kontener Amazon S3. Jeśli zdecydujesz się przechowywać wiadomości kanałowe w Amazon. W przypadku zarządzanego kontenera S3 musisz przyznać usłudze AWS IoT Analytics uprawnienie do wykonywania tych działań w Twoim imieniu na Twoim koncie **Amazon S3: s3:GetBucketLocation** (weryfikacja lokalizacji konta) **s3:PutObject** (przechowywanie), **s3:GetObject** (odczyt), **s3:ListBucket** (ponowne przetwarzanie).

Przykład:



```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-iot-analytics-bucket",
        "arn:aws:s3:::my-iot-analytics-bucket/*"
      ]
    }
  ]
}
```

Rysunek 9 Przyznanie usługom uprawnień

4.5.3 Tworzenie magazynu danych

Magazyn danych odbiera i przechowuje Twoje wiadomości. Nie jest to baza danych, ale skalowalne i możliwe do przeszukiwania repozytorium Twoich wiadomości. Możesz utworzyć wiele magazynów danych, aby przechowywać wiadomości pochodzące z różnych urządzeń lub lokalizacji, lub możesz użyć jednego magazynu danych, aby odbierać wszystkie wiadomości AWS IoT.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

Aby uzyskać więcej informacji o magazynie danych.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

Aby wyświetlić listę magazynów danych, które już utworzono.

```
aws iotanalytics list-datastores
```

4.6 Microsoft Azure IoT Hub:

Zalety:

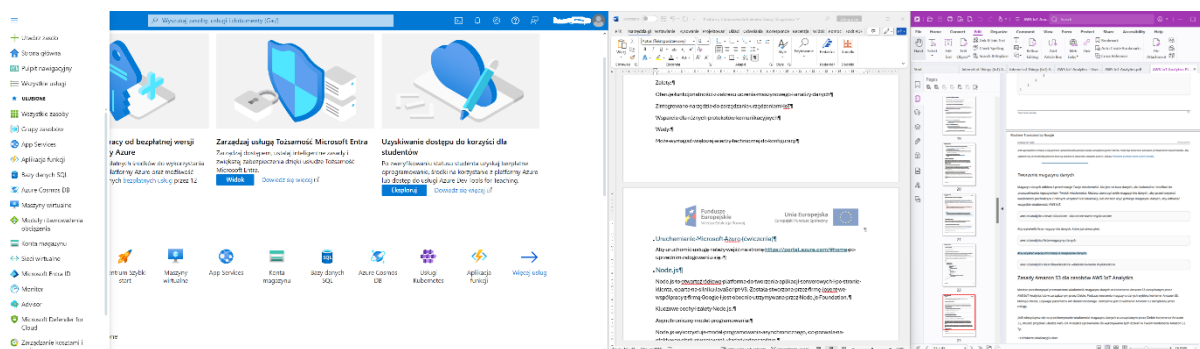
- Oferuje funkcjonalności z zakresu uczenia maszynowego i analizy danych
- Zintegrowane narzędzia do zarządzania urządzeniami IoT
- Wsparcie dla różnych protokołów komunikacyjnych

Wady:

- Może wymagać większej wiedzy technicznej do konfiguracji

4.6.1 Uruchamianie Microsoft Azure (ćwiczenia)

Aby uruchomić usługę należy wejść na stronę <https://portal.azure.com/#home> po uprzednim zalogowaniu się. Rysunek poniżej przedstawia stronę główną po zalogowaniu. W rozwijalnym menu po lewej można przejść do poszczególnych zasobów.



Rysunek 10 Panel klienta Azure <https://portal.azure.com>

4.6.2 Wysyłanie danych telemetrycznych z urządzenia do centrum IoT (ćwiczenia)

Uruchamiamy Cloud Shell poprzez wybranie przycisku na pasku menu.



Rysunek 11 Pasek menu witryny Azure Portal



4.6.2.1 Wybieramy środowisko CLI²⁰ do wyboru Bash²¹ lub PowerShell

22



4.6.2.2 Przygotowanie dwóch sesji Azure CLI.

Jeśli używasz Cloud Shell, uruchomisz te sesje na oddzielnych kartach Cloud Shell. Jeśli używasz lokalnego klienta CLI, uruchomisz oddzielne wystąpienia CLI. Użyj oddzielnych sesji CLI do następujących zadań:

Pierwsza sesja symuluje urządzenie IoT komunikujące się z centralą IoT. Druga sesja albo monitoruje urządzenie z pierwszej sesji, albo wysyła wiadomości, polecenia i aktualizacje właściwości.

Aby uruchomić polecenie, wybierz opcję **Copy**, aby skopiować blok kodu z tego szybkiego startu, wkleić go do sesji powłoki i uruchomić.

Azure CLI wymaga zalogowania się do konta Azure. Cała komunikacja między sesją powłoki Azure CLI a centrum IoT jest uwierzytelniana i szyfrowana. W rezultacie ten szybki start nie wymaga dodatkowego uwierzytelniania, którego użyłbyś w przypadku prawdziwego urządzenia, takiego jak ciąg połączenia.

W pierwszej sesji CLI uruchom polecenie **az extension add**. Polecenie dodaje rozszerzenie Microsoft Azure IoT Extension for Azure CLI do powłoki CLI. Rozszerzenie IOT Extension dodaje polecenia specyficzne dla IoT Hub, IoT Edge i IoT Device Provisioning Service (DPS) do Azure CLI.

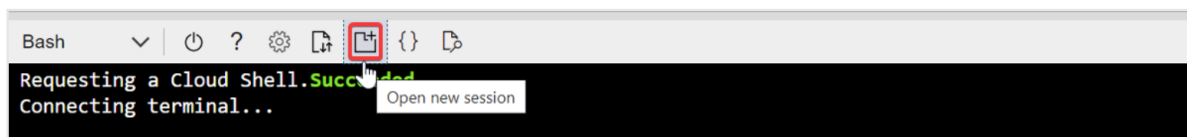
```
az extension add --name azure-iot
```

²⁰ Środowisko CLI (Command Line Interface) to interfejs wiersza poleceń, który pozwala użytkownikom na interakcję z systemem operacyjnym lub aplikacjami za pomocą tekstowych poleceń wprowadzanych z klawiatury

²¹ Bash (Bourne Again SHell) to popularny interfejs wiersza poleceń dla systemów uniksowych, takich jak Linux i macOS.

²² PowerShell to zaawansowany interfejs wiersza poleceń i język skryptowy stworzony przez Microsoft, który zapewnia nowoczesne i elastyczne środowisko do zarządzania systemami Windows i automatyzacji zadań.

Po zainstalowaniu rozszerzenia Azure IOT nie trzeba instalować go ponownie w żadnej sesji Cloud Shell. Następnie otwórz drugą sesję CLI. Jeśli używasz Cloud Shell w przeglądarce, użyj przycisku **Open new session**. Jeśli używasz CLI lokalnie, otwórz drugą instancję CLI.



4.6.2.3 Tworzenie centrum IoT

Wykorzystamy interfejs wiersza poleceń platformy Azure, aby utworzyć grupę zasobów i centrum IoT. Grupa zasobów platformy Azure to logiczny kontener, w którym wdrażane i zarządzane są zasoby platformy Azure. Centrum IoT działa jako centralne centrum komunikatów do dwukierunkowej komunikacji między aplikacją IoT a urządzeniami.

W pierwszej sesji CLI uruchom polecenie **az group create**, aby utworzyć grupę zasobów. Poniższe polecenie tworzy grupę zasobów o nazwie *MyResourceGroup* w lokalizacji *eastus*.

```
az group create --name MyResourceGroup --location eastus
```

Następnie uruchom polecenie **Az PowerShell module iot hub create**, aby utworzyć centrum IoT. Utworzenie centrum IoT zajmuje kilka minut. *YourIoTHubName* . Zastąp ten symbol zastępczy i otaczające go nawiasy klamrowe w poniższym poleceniu, używając nazwy wybranej dla swojego centrum IoT. Nazwa centrum IoT musi być globalnie unikatowa w usłudze Azure. Używaj swojej nazwy centrum IoT w pozostałej części tego szybkiego startu, gdziekolwiek zobaczysz symbol zastępczy.

```
az iot hub create --resource-group MyResourceGroup --name {YourIoTHubName}
```

Kolejne polecenie **az iot device simulate** . To polecenie uruchamia symulowane urządzenie. Urządzenie wysyła dane telemetryczne do Twojego centrum IoT i odbiera z niego wiadomości. *YourIoTHubName* . Zastąp ten symbol zastępczy poniżej nazwą wybraną dla swojego centrum IoT.

```
az iot device simulate -d simDevice -n {YourIoTHubName}
```



Aby monitorować urządzenie:

W drugiej sesji CLI uruchom polecenie **az iot hub monitor-events**. To polecenie stale monitoruje symulowane urządzenie. Dane wyjściowe pokazują dane telemetryczne, takie jak zdarzenia i zmiany stanu właściwości, które symulowane urządzenie wysyła do centrum IoT. *YourIoTHubName* zastępujemy nazwą wybraną dla swojego centrum IoT.

```
az iot hub monitor-events --output table -p all -n {YourIoTHubName}
```

```
Bash
Starting event monitor, use ctrl-c to stop...
event:
  annotations:
    iothub-connection-auth-generation-id:
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: simDevice
    iothub-enqueuedtime: 1653493361818
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1653493361826
    x-opt-offset: '59904'
    x-opt-sequence-number: 117
  component: ''
  interface: ''
  module: ''
  origin: simDevice
  payload:
    data: 'Ping from Az CLI IoT Extension #14'
    id:
    timestamp: '2022-05-25 15:42:41.809391'
  properties:
    application: {}
    system:
      content_encoding: utf-8
      content_type: application/json
```

Rysunek 12 Monitorowanie urządzenia Azure

Po monitorowaniu symulowanego urządzenia w drugiej sesji CLI naciśnij Ctrl+C, aby zatrzymać monitorowanie. Pozostaw drugą sesję CLI otwartą, aby użyć jej w późniejszych krokach.

4.6.2.4 Wysyłanie wiadomości

W tej sekcji wysyłasz wiadomość do symulowanego urządzenia.

W pierwszej sesji CLI potwierdź, że symulowane urządzenie nadal działa. Jeśli urządzenie się zatrzymało, uruchom następujące polecenie, aby je ponownie uruchomić: Oczywiście *YourIoTHubName* zastępujesz swoją nazwą.

```
az iot device simulate -d simDevice -n {YourIoTHubName}
```

W drugiej sesji CLI uruchom polecenie **az iot device c2d-message send** . To polecenie wysła komunikat **cloud-to-device** z Twojego centrum IoT do symulowanego urządzenia. Komunikat zawiera ciąg i dwie pary klucz-wartość.

```
az iot hub invoke-device-method --mn MySampleMethod -d simDevice -n {YourIoTHubName}
```

W pierwszej sesji CLI sprawdź, czy dane wyjściowe pokazują wywołanie metody.

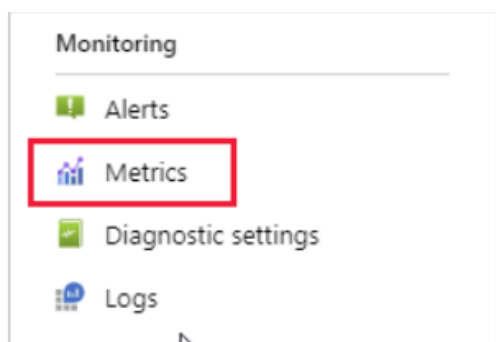
```
{
  "payload": {
    "methodName": "MySampleMethod",
    "methodRequestId": "1",
    "methodRequestPayload": {}
  },
  "status": 200
}
```

4.6.2.5 Przeglądanie metryki wiadomości w portalu

Portal Azure umożliwia zarządzanie wszystkimi aspektami centrum IoT i urządzeń. W typowej aplikacji centrum IoT, która pobiera dane telemetryczne z urządzeń, możesz chcieć monitorować urządzenia lub wyświetlać metryki dotyczące danych telemetrycznych urządzeń.

Aby wizualizować metryki wiadomości w witrynie Azure Portal należy:

W lewym menu nawigacyjnym portalu wybrać opcję Wszystkie zasoby. Ta karta zawiera listę wszystkich zasobów w Twojej subskrypcji, w tym utworzonego przez Ciebie centrum IoT. Wybierz łącze w utworzonym przez siebie centrum IoT. Portal wyświetla stronę przeglądu dla centrum. Wybierz opcję Metryki w lewym panelu IoT Hub.



Rysunek 13 Wybieranie okna metryki

Następnie w polu W polu Zakres wprowadź nazwę swojego centrum IoT.

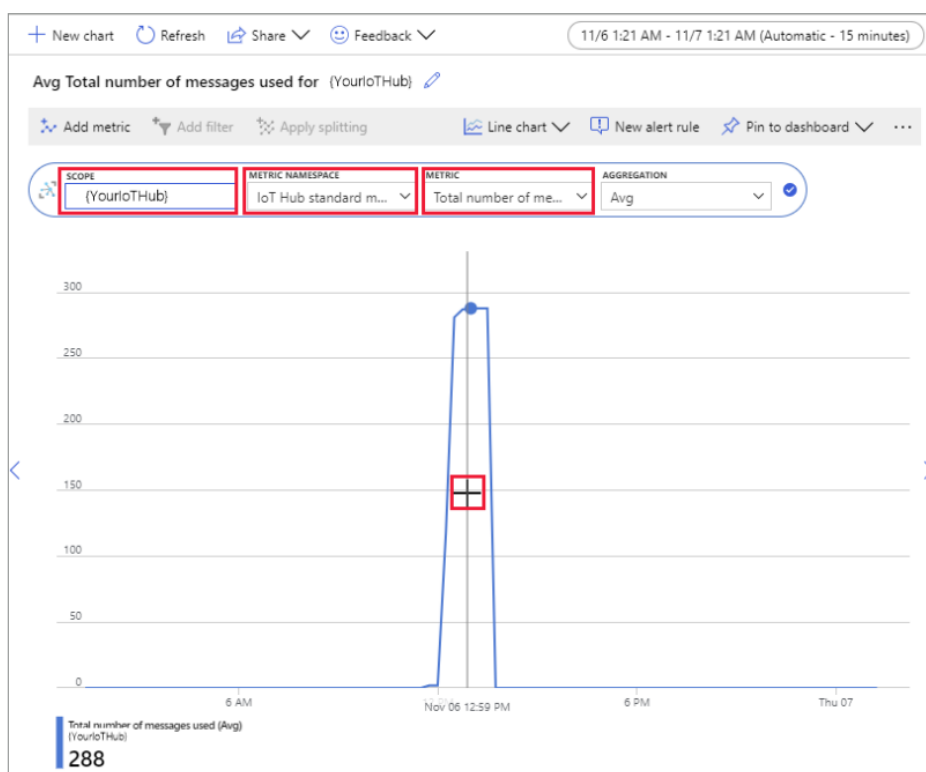


W polu Przestrzeń nazw metryk wybierz opcję Standardowe metryki Iot Hub.

W polu Metryka wybierz opcję Całkowita liczba użytych wiadomości.

Najedź kursorem myszy na obszar osi czasu, w którym urządzenie wysłało wiadomości.

Całkowita liczba wiadomości w danym momencie pojawia się w lewym dolnym rogu osi czasu.



Rysunek 14 Całkowita liczba wiadomości

Opcjonalnie użyj rozwijanej listy **Metric**, aby wyświetlić inne metryki na symulowanym urządzeniu. Na przykład, Ukończone dostawy wiadomości (ang. *C2d message deliveries completed*) lub Łączna liczba urządzeń (podgląd) (ang. *Total devices (preview)*).

4.7 IBM Watson IoT

IBM Watson IoT²³ to platforma do zarządzania Internetem Rzeczy (IoT) stworzona przez firmę IBM. Jest to kompleksowe rozwiązanie, które umożliwia zbieranie, przetwarzanie i analizowanie danych z urządzeń IoT w celu uzyskania wglądów i optymalizacji procesów.

Kluczowe cechy i funkcje IBM Watson IoT:

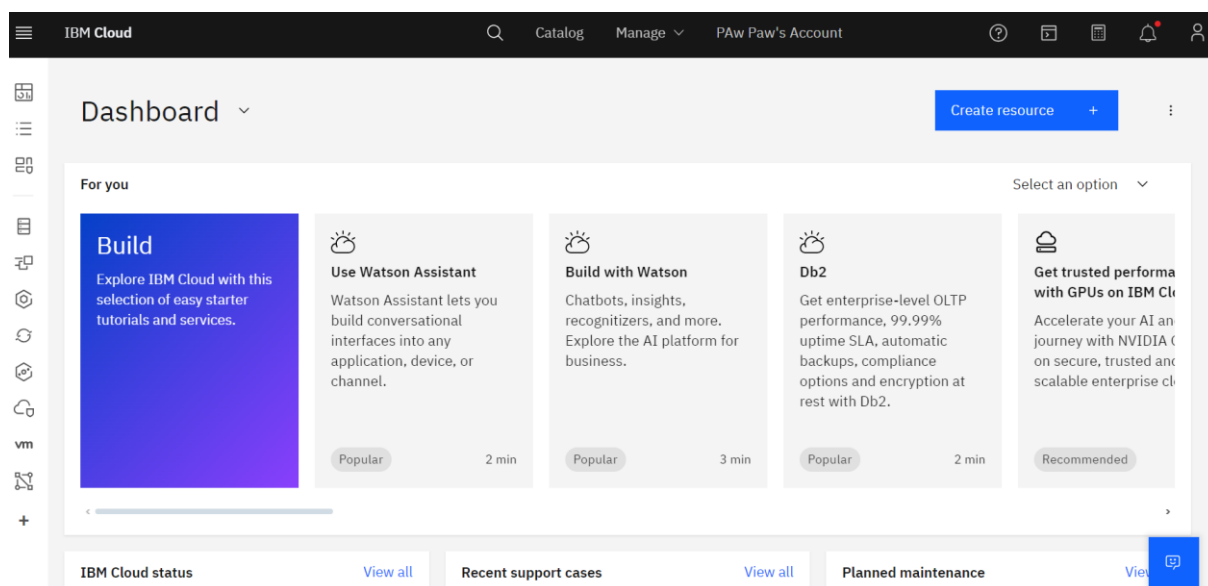
²³ www.ibm.com/cloud/internet-of-things



- Zarządzanie urządzeniami: Platforma umożliwia rejestrowanie, zarządzanie i monitorowanie urządzeń IoT.
- Zbieranie danych: IBM Watson IoT może zbierać dane z różnych źródeł, takich jak czujniki, urządzenia mobilne i inne urządzenia IoT.
- Przetwarzanie danych: Platforma oferuje możliwości przetwarzania danych w czasie rzeczywistym oraz przetwarzania historycznego.
- Analiza danych: IBM Watson IoT wykorzystuje zaawansowane algorytmy uczenia maszynowego do analizy danych i wykrywania wzorców.
- Wizualizacja danych: Platforma umożliwia tworzenie dashboardów i raportów, które ułatwiają wizualizację i interpretację danych.
- Integracja z innymi usługami IBM: IBM Watson IoT może być zintegrowany z innymi usługami chmurowymi IBM, takimi jak IBM Watson Studio, IBM Watson Machine Learning, IBM Cloud Object Storage i inne.
- Bezpieczeństwo: Platforma oferuje zaawansowane funkcje bezpieczeństwa, w tym szyfrowanie danych, kontrolę dostępu i zarządzanie tożsamościami.
- Skalowalność: IBM Watson IoT jest zaprojektowany do obsługi dużej liczby urządzeń i generowanych przez nie danych.
- Zgodność z przepisami: Platforma jest zgodna z wieloma przepisami dotyczącymi ochrony danych, takimi jak GDPR i HIPAA.
- Interoperacyjność: IBM Watson IoT wspiera różne protokoły komunikacyjne, takie jak MQTT, HTTP i AMQP, co umożliwia integrację z różnymi systemami i urządzeniami.

IBM Watson IoT jest wykorzystywany w wielu branżach, takich jak produkcja, transport, energetyka, opieka zdrowotna i inne, do zarządzania flotą pojazdów, monitorowania stanu maszyn, optymalizacji zużycia energii i wielu innych zastosowań.

Na rysunku poniżej widzimy panel IBM Cloud do którego wchodzimy po zalogowaniu na stronie **<https://cloud.ibm.com/login>**



Rysunek 15 Panel klienta IBM Cloud

Instalacja IBM Watson IoT Platform - Message Gateway obejmuje kilka kluczowych kroków:

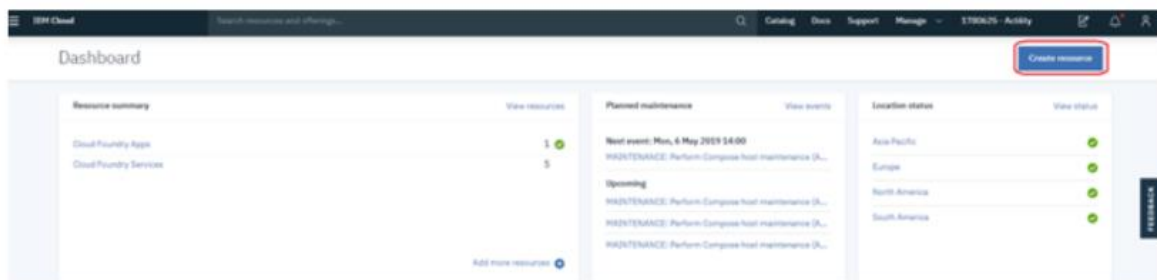
- Przygotowanie systemu operacyjnego: Zainstaluj odpowiedni system operacyjny (np. Red Hat, CentOS). Jeśli instalujesz w kontenerze Docker, zainstaluj Docker na swoim systemie operacyjnym.
- Pobranie i przygotowanie pakietów instalacyjnych: Pobierz pliki .tz, które zawierają wszystkie niezbędne pliki do instalacji.
- Instalacja IBM Watson IoT Platform - Message Gateway: Możesz zainstalować platformę w kontenerach Docker lub bezpośrednio na swoim systemie operacyjnym jako RPM. Na systemach Red Hat potrzebujesz aktywnej subskrypcji Red Hat do instalacji w kontenerach Docker lub jako RPM. Na systemach CentOS nie jest to wymagane. Przyjmij warunki licencji podczas instalacji.
- Logowanie do interfejsu webowego: Po zakończeniu instalacji zaloguj się do interfejsu webowego IBM Watson IoT Platform - Message Gateway, aby zweryfikować poprawność instalacji i zarządzać serwerem komunikacyjnym.
- Weryfikacja instalacji: Możesz użyć aplikacji Messaging Tester, aby przetestować i zweryfikować poprawność instalacji.

Ten proces instalacji zapewnia, że IBM Watson IoT Platform - Message Gateway jest prawidłowo skonfigurowany i gotowy do użycia w Twojej infrastrukturze IoT.

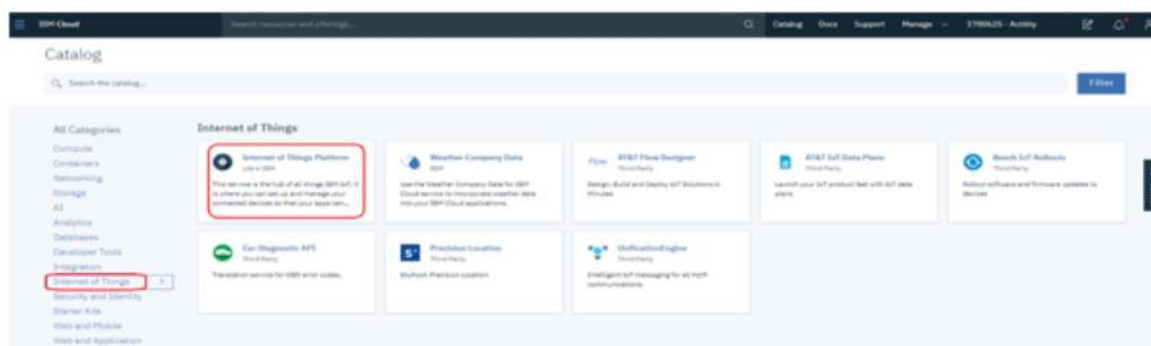


Przykłady kodu i szablony pod adresem <https://github.com/ibm-watson-iot>

Utworzyć platformę IBM Watson IoT możemy również z panelu IBM Cloud wybierając Create resource jak na rysunku poniżej:



Następnie wybieramy platformę **Internet Of Things**:



Wypełniamy poszczególne okna i wybieramy plan Lite/free bez opłat



This service is the hub for IBM Watson IoT and lets you communicate with and consume data from connected devices and gateways. Use the built-in web console dashboards to monitor your IoT data and analyze it in real time. Then, enhance and customize your IBM Watson IoT Platform experience by building and connecting your own apps by using messaging and REST APIs.

[View Docs](#) [Terms](#)

AUTHOR IBM
PUBLISHED 19/02/2019
TYPE Service
LOCATION Frankfurt, London, Dallas

Service name: **activityIoDemoPlatform**

Choose a region/location to deploy in: **Frankfurt**

Choose an organization: **activityDemoOrg**

Choose a space: **activityDemoSpace**

Tags: [Examples: emvdes, version:1](#)

Features

- Connect**
Quickly and securely register and connect your devices and gateways. You can find simple step-by-step instructions for connecting popular devices, sensors, and gateways in our recipes site.
- Analyze in real time**
Monitor your real-time device data through rules, analytics, and dashboards. Define rules to monitor conditions and trigger automatic actions that include alerts, email, DTTT, Node RED flows, and external services to react quickly to critical changes.
- Information Management**
Control what happens to the data that is received from your connected devices. Manage data storage, configure data transformation actions, and integrate with other data services and device platforms.
- Risk and Security management**
Our secure-by-design control capabilities protect the integrity of your IoT solution through secure connectivity and access control for users and applications. Extend the base security with threat intelligence for IoT to visualize critical risks and automate operational responses with policy-driven mitigation actions.

Images

Images can be screen captures, slides, or videos. Click an image to view the details.

Pricing Plans

Monthly prices shown are for country or region: France

PLAN	FEATURES	PRICING
☒ Lite	Includes up to 500 registered devices, and a maximum of 200 MB of each data metric. Maximum of 500 registered devices. Maximum of 500 application bindings. Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed. The Lite service plan for Internet of Things Platform includes up to 500 registered devices, and a maximum of 200 MB each of data exchanged, data analyzed, and edge data analyzed per month. Lite plan services are deleted after 30 days of inactivity.	Free
Watson IoT Platform Connection and Analytics Service Capacity Level 1	IBM Watson IoT Platform (Connection Service and Analytics Service) is a ready-to-run, pre-integrated SaaS managed service IoT platform with capabilities in connectivity, data management and advanced analytics. Capacity Level 1 includes 5,000 sensor devices and 1 million analytics events.	€1,885.00 EUR/Plan
Watson IoT Platform Connection and Analytics Service Capacity Level 2	IBM Watson IoT Platform (Connection Service and Analytics Service) is a ready-to-run, pre-integrated SaaS managed service IoT platform with capabilities in connectivity, data management and advanced analytics. Capacity Level 2 includes 25,000 sensor devices and 5 million analytics events.	€4,018.00 EUR/Plan

[Need Help? Contact IBM Cloud Support](#)

[Add to estimate](#) [Create](#)

Na następnej stronie klikamy **połącz** aby uruchomić platformę IoT:

IBM Cloud

Search resources and offerings...

Manage Plan Connections

Resource list

activityIoDemoPlatform
Location: Frankfurt Org: activityDemoOrg Space: activityDemoSpace [Add Tags](#)

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

[Launch](#) [Docs](#)

Ready for the next level?

IBM Watson IoT Platform Journey

☒ **Lite**

The Lite service plan provides a lightweight development environment to get you started with the connectivity capabilities of Watson IoT Platform.

- Free
- 200 MB data-transfer limit
- 500 application bindings limit
- 500 registered devices limit

☐ **Non-Production**

The Non-Production service plan is a full-featured, fully-integrated offering that enables you to explore Watson IoT Platform to see how the service can fit into your IoT environment.

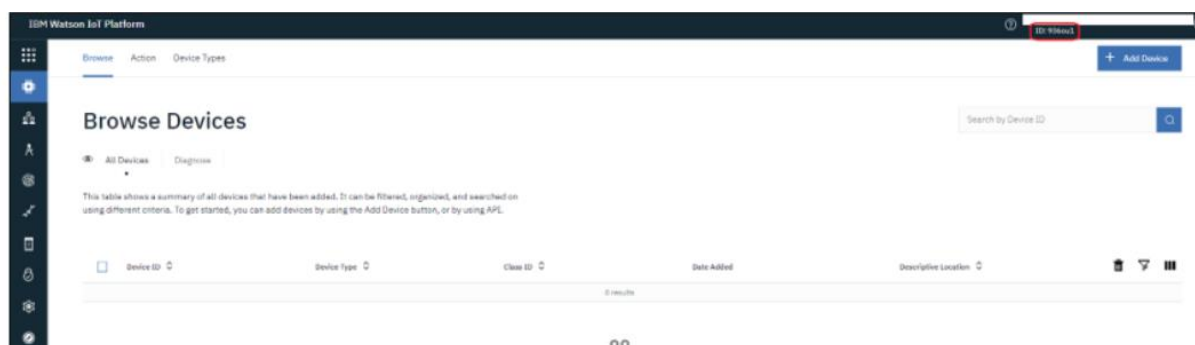
- Starts at \$500 per month
- Capacity limit based on device type
- Optional Analytics Service and Blockchain Service add-ons

☐ **Production**

The Production service is a fully managed SaaS offering that enables you to manage and analyze enterprise IoT data.

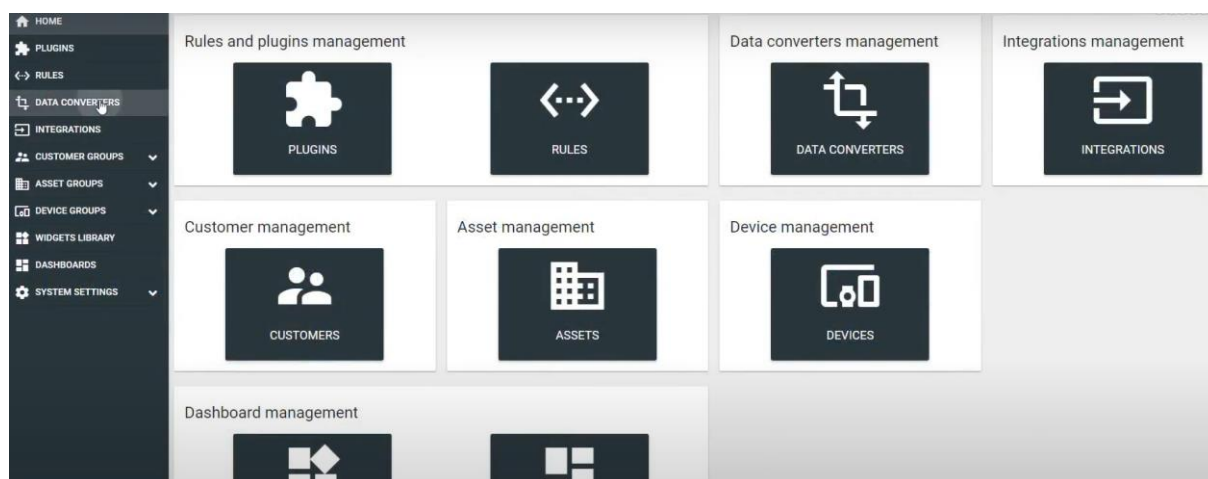
- Includes IBM Service & Support
- Pricing based on number of devices per device type
- Optional Analytics Service and Blockchain Service add-ons

Zostaniemy przekierowani do panelu stworzonej przez nas platformy IBM Watson IoT:



4.7.1 Konfiguracja IBM Watson IoT (ćwiczenia)

Wybieramy z listy po lewej Data Converters jak na rysunku poniżej



Rysunek 16 Panel IBM Watson IoT

Protokół MQTT over TLS jest używany przez IBM Watson IoT do komunikacji między urządzeniami a chmurą. Aby skonfigurować połączenie, należy wykonać następujące kroki:

- Utwórz połączenie: Przed utworzeniem przepływu danych (flow) musisz utworzyć połączenie, które nawiąże łączność między urządzeniem a chmurą IBM Watson IoT.
- Endpointy API: ²⁴Aby utworzyć, zmodyfikować lub usunąć połączenie, użyj następujących endpointów API:
 - **POST /connections** - do tworzenia połączenia
 - **PUT /connections** - do modyfikacji połączenia
 - **DELETE /connections** - do usunięcia połączenia

²⁴ Endpoint API to adres URL, który umożliwia komunikację z usługą lub systemem za pośrednictwem protokołu HTTP. W kontekście IBM Watson IoT, endpointy API są używane do wykonywania różnych operacji na platformie, takich jak tworzenie, modyfikowanie i usuwanie połączeń, zarządzanie urządzeniami, przysyłanie danych i wiele innych.



- Zastosowanie połączenia: Połączenie może być używane do transportowania dowolnego przepływu danych (Uplink), niezależnie od parametru DevEUI urządzenia.
- Konfiguracja: Aby skonfigurować połączenie, musisz podać odpowiednie informacje, takie jak identyfikator urządzenia (DevEUI), klucz urządzenia (DevKey) oraz inne wymagane dane. Bezpieczeństwo: Protokół MQTT over TLS zapewnia bezpieczną komunikację między urządzeniem a chmurą, co jest kluczowe dla ochrony danych i zgodności z przepisami dotyczącymi bezpieczeństwa.
- Pamiętaj, że po utworzeniu połączenia możesz utworzyć przepływ danych (flow), który będzie odpowiedzialny za przesyłanie danych z urządzenia do chmury IBM Watson IoT.

Przykład stworzonego połączenia:

```
POST /connections

{
  "connectorId": "actility-watson-iot",
  "name": "Watson Demo Connection",
  "configuration": {
    "description": "Watson Datacenter",
    "orgId": "936ou1",
    "deviceType": "actilitydevicetype",
    "gatewayType": "actilitygatewaytype",
    "gatewayId": "actilitygateway",
    "gatewayToken": "GXvftVxJXoI@Z(ufxU",
    "apiKey": "a-936ou1-4ocurmbuva",
    "apiToken": "Sgbw?FYqeK6&sIhDoe"
  }
}
```

Rysunek 17 Połączenie z Watson IoT

W poniższej tabeli wymieniono oczekiwane rezultaty właściwości po zastosowaniu:

Tabela 2 Lista właściwości i oczekiwanych rezultatów

Właściwość (<i>ang. property</i>)	Oczekiwane rezultaty (<i>ang. Expected results</i>)
connectorId	Łączenie z Watson IoT (<i>ang. Set to actility-watson-iot</i>)
configuration/orgId	Identyfikator organizacji konta Watson IoT (<i>ang. Organization id of the Watson IoT account.</i>)
configuration/gatewayId	Identyfikator bramy Watson IoT używanej przez złącze. (<i>ang. Id of the Watson IoT gateway used by the connector.)</i>



configuration/gatewayType	Typ bramy Watson IoT bramy Watson IoT używanej przez złącze (<i>ang. Watson IoT gateway type of the Watson IoT gateway used by the connector</i>).
configuration/gatewayToken	Token uwierzytelniający bramy Watson IoT używany przez łącznik. (<i>ang. Authentication token of the Watson IoT gateway used by the connector. </i>)
configuration/deviceType	Typ urządzenia Watson IoT, który ma zostać powiązany z urządzeniami przetwarzanymi przez złącze. (<i>ang. Watson IoT Device type to be associated with the Devices processed by the connector</i>).
configuration/apiKey	Klucz API aplikacji Watson IoT, który ma zostać powiązany ze złączem. (<i>ang. API Key of the Watson IoT application to be associated with the connector. </i>)
configuration/apiToken	Token uwierzytelniający aplikacji Watson IoT, który ma być powiązany z łącznikiem. (<i>ang. Authentication Token of the Watson IoT application to be associated with the connector</i>).

4.8 Node.js

Node.js to platforma Open Source do tworzenia aplikacji serwerowych i po stronie klienta, oparta na silniku JavaScript V8. Została stworzona przez firmę Joyent we współpracy z firmą Google i jest obecnie utrzymywana przez Node.js Foundation²⁵.

Kluczowe cechy i zalety Node.js:

- Asynchroniczny model programowania: Node.js wykorzystuje model programowania asynchronicznego, co pozwala na efektywne obsługiwanie wielu żądań jednocześnie. Umożliwia to tworzenie wydajnych aplikacji serwerowych, które mogą obsługiwać dużą liczbę połączeń.
- Jednowątkowość: Node.js działa w jednym wątku, co upraszcza zarządzanie zasobami i eliminuje problemy związane z synchronizacją wątków. Wykorzystuje mechanizmy I/O (wejścia/wyjścia) asynchroniczne, co pozwala na efektywne wykorzystanie zasobów.
- Ekosystem npm: Node.js korzysta z repozytorium pakietów npm (Node Package Manager), które zawiera ogromną liczbę gotowych bibliotek i modułów. Ułatwia to rozszerzanie funkcjonalności aplikacji i korzystanie z istniejących rozwiązań.

²⁵ <https://openjsf.org>



- Wsparcie dla różnych frameworków²⁶: Node.js wspiera wiele popularnych frameworków do tworzenia aplikacji internetowych, takich jak Express.js, Hapi.js, Koa.js i inne. Frameworki te ułatwiają tworzenie i zarządzanie aplikacjami.
- Łatwość integracji z innymi technologiami: Node.js może łatwo integrować się z innymi technologiami, takimi jak bazy danych (MongoDB, PostgreSQL, MySQL), systemy plików (S3, SMB), usługi w chmurze (AWS, Azure) i wiele innych. Umożliwia to tworzenie złożonych aplikacji korzystających z różnych usług.
- Wsparcie dla różnych środowisk: Node.js może być uruchamiany w różnych środowiskach, takich jak serwery, urządzenia IoT, systemy wbudowane i inne. Umożliwia to tworzenie aplikacji działających na różnych platformach.
- Wsparcie dla TypeScript: Node.js wspiera TypeScript, co pozwala na dodanie statycznego typowania do kodu JavaScript. Ułatwia to tworzenie bardziej złożonych aplikacji i zwiększa ich czytelność.
- Wsparcie dla różnych języków: Oprócz JavaScript, Node.js wspiera również inne języki, takie jak TypeScript, CoffeeScript i inne. Umożliwia to wybór języka, który najlepiej pasuje do danego projektu.

Node.js jest popularnym wyborem do tworzenia aplikacji serwerowych, szczególnie w przypadku aplikacji wymagających wysokiej wydajności i obsługi dużej liczby połączeń. Jego asynchroniczny model programowania i bogaty ekosystem pakietów sprawiają, że jest łatwy w użyciu i skalowalny. Jednak ze względu na jednowątkowość, może nie być odpowiedni dla zadań wymagających intensywnych obliczeń CPU. Warto również zauważyć, że Node.js jest często wykorzystywany do tworzenia aplikacji backendowych, ale może być również używany do tworzenia aplikacji po stronie klienta, takich jak aplikacje desktopowe lub aplikacje mobilne.

Node.js wykorzystuje model programowania asynchronicznego, co pozwala na efektywne obsługiwanie operacji wejścia/wyjścia, takich jak odczyt z sieci, dostęp do bazy danych lub systemu plików. Zamiast blokować wątek i marnować cykle procesora na oczekiwanie, Node.js wznawia operacje po otrzymaniu odpowiedzi. Dzięki temu może obsługiwać tysiące

²⁶ Framework (w języku polskim często tłumaczony jako szkielet lub struktura) to zestaw narzędzi, bibliotek, wzorców projektowych i konwencji, które ułatwiają tworzenie, rozwijanie i utrzymywanie oprogramowania. Frameworki są przeznaczone do rozwiązywania konkretnych problemów lub zadań w określonej dziedzinie, takiej jak tworzenie aplikacji webowych, aplikacji mobilnych, systemów zarządzania bazami danych itp.



równoczesnych połączeń z jednym serwerem bez konieczności zarządzania współbieżnością wątków, co mogłoby być źródłem błędów.

Jedną z głównych zalet Node.js jest to, że miliony programistów front-end, którzy piszą JavaScript dla przeglądarek, mogą teraz tworzyć kod po stronie serwera bez konieczności nauki zupełnie innego języka. Wykorzystując ten sam język JavaScript, mogą łatwo przenosić swoje umiejętności i wiedzę między kodem po stronie klienta a kodem po stronie serwera.

Ponadto, Node.js umożliwia wykorzystanie nowych standardów ECMAScript bez konieczności czekania, aż wszyscy użytkownicy zaktualizują swoje przeglądarki. Użytkownik może samodzielnie zdecydować, której wersji ECMAScript użyć, zmieniając wersję Node.js. Dodatkowo, można włączać określone funkcje eksperymentalne, uruchamiając Node.js z odpowiednimi flagami.

Dzięki tym cechom, Node.js stanowi wyjątkowo atrakcyjną platformę do tworzenia wydajnych aplikacji serwerowych, szczególnie dla programistów JavaScript z doświadczeniem w tworzeniu kodu po stronie klienta.

4.8.1 Przykład aplikacji Node.js

Poniżej przykład popularnej aplikacji „Hello Word” jako serwer WWW.

```
CJS  MJS

1  const { createServer } = require('node:http');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  const server = createServer((req, res) => {
7    res.statusCode = 200;
8    res.setHeader('Content-Type', 'text/plain');
9    res.end('Hello World');
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`Server running at http://${hostname}:${port}/`);
14 });
```

JavaScript

Copy to clipboard

Rysunek 18 Hello Word w Node.js



Aby uruchomić podany fragment kodu, wykonaj następujące kroki:

- Zapisz kod jako plik:
- Jeśli kod jest napisany w formacie .js, zapisz go jako **server.js**.
- Jeśli kod jest napisany w formacie .mjs, zapisz go jako **server.mjs**.
- Uruchom kod w terminalu:
- Użyj polecenia **node** w terminalu, aby uruchomić plik.
- Jeśli plik ma rozszerzenie .js, użyj polecenia:
node server.js
- Jeśli plik ma rozszerzenie .mjs, użyj polecenia:
node server.mjs

Kod, który został przedstawiony, wykorzystuje moduł http z biblioteki standardowej Node.js do utworzenia prostego serwera HTTP. Oto szczegółowe wyjaśnienie:

Importowanie modułu http:

```
const http = require('http');
```

Moduł http zapewnia funkcjonalność do tworzenia i obsługi serwerów HTTP.

Tworzenie serwera HTTP:

```
const server = http.createServer((req, res) => {  
  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  
  res.end('Hello World\n');  
  
});
```

Metoda **createServer()** tworzy nowy serwer HTTP i zwraca go. Funkcja wywołania zwrotnego (callback) jest wykonywana za każdym razem, gdy serwer otrzyma nowe żądanie.

Ustawianie portu i hosta:

```
server.listen(8080, '127.0.0.1', () => {  
  
  console.log('Server running at http://127.0.0.1:8080/');  
  
});
```



Metoda **listen()** ustawia port (8080) i adres IP (127.0.0.1), na którym serwer będzie nasłuchiwał. Funkcja wywołania zwrotnego jest wykonywana, gdy serwer jest gotowy do przyjmowania żądań.

Obsługa żądań:

```
const server = http.createServer((req, res) => {  
  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  
  res.end('Hello World\n');  
  
});
```

Funkcja wywołania zwrotnego (**callback**) otrzymuje dwa obiekty: **req** (żądanie) i **res** (odpowiedź). Za każdym razem, gdy serwer otrzyma nowe żądanie, ta funkcja jest wykonywana.

req (obiekt http.IncomingMessage): Zawiera szczegóły żądania HTTP, takie jak nagłówki, metoda HTTP, ścieżka URL, dane żądania itp.

res (obiekt http.ServerResponse): Służy do wysyłania odpowiedzi HTTP do klienta.

W tym prostym przykładzie, serwer odpowiada na każde żądanie z kodem statusu 200 (OK) i wysyła prostą odpowiedź tekstową "Hello World".

Aby przetestować działanie serwera, otwórz przeglądarkę i przejdź do adresu **http://127.0.0.1:8080/**. Powinieneś zobaczyć wiadomość "Hello World" w przeglądarce.

4.8.2 Instalacja Node.js

System Windows. Należy uruchomić PowerShell i wykonać komendy jak na rysunku poniżej.



```
1 # installs fnm (Fast Node Manager)
2 winget install Schniz.fnm
3
4 # configure fnm environment
5 fnm env --use-on-cd | Out-String | Invoke-Expression
6
7 # download and install Node.js
8 fnm use --install-if-missing 20
9
10 # verifies the right Node.js version is in the environment
11 node -v # should print `v20.17.0`
12
13 # verifies the right npm version is in the environment
14 npm -v # should print `10.8.2`
```

PowerShell

 Copy to clipboard

Rysunek 19 Instalacja Node.js Windows

System Linux i MacOS. Komendy na rysunku poniżej:

```
1 # installs fnm (Fast Node Manager)
2 curl -fsSL https://fnm.vercel.app/install | bash
3
4 # activate fnm
5 source ~/.bashrc
6
7 # download and install Node.js
8 fnm use --install-if-missing 20
9
10 # verifies the right Node.js version is in the environment
11 node -v # should print `v20.17.0`
12
13 # verifies the right npm version is in the environment
14 npm -v # should print `10.8.2`
```

Bash

 Copy to clipboard

Uruchomienie programu Hello Word

Tworzymy folder o nazwie **Projekt**. W notatniku lub innym edytorze tekstu tworzymy plik o nazwie **HelloWord** i kopiujemy do niego tekst z rysunku poniżej



```
const http = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Rysunek 20 Plik HelloWorld.js

Zapisujemy plik, a następnie otwieramy terminal windows i uruchamiamy stworzony plik

```
node hello-world.js
```

Wyjście kierujemy na serwer pod adresem:

```
Server running at http://127.0.0.1:3000/
```

Otwieramy przeglądarkę internetową i wchodzimy na adres <http://127.0.0.1:3000>

Wyświetli się napis **Hello, Word!** co oznacza, że serwer został uruchomiony.

4.9 Wykorzystanie Node.js do implementacji warstwy WWW z czujnikami

Poniżej poszczególne kroki

4.9.1 Wybór odpowiedniego frameworka:

Express.js²⁷: Jest to popularny framework do tworzenia aplikacji webowych w Node.js. Umożliwia łatwe definiowanie tras, obsługę żądań HTTP i renderowanie widoków. Jest to darmowe oprogramowanie o otwartym kodzie źródłowym wydane na licencji MIT. Jest przeznaczony do tworzenia aplikacji internetowych, aplikacji mobilnych i API. Express jest minimalny i elastyczny, zapewniając solidny zestaw funkcji dla aplikacji internetowych i

²⁷ <https://expressjs.com>



mobilnych. Filozofia Express polega na zapewnieniu małych, solidnych narzędzi dla serwerów HTTP, co czyni go doskonałym rozwiązaniem dla aplikacji jednostronicowych, stron internetowych, hybryd lub API.

Koa.js²⁸: Alternatywa dla Express.js, która oferuje większą elastyczność i lepszą wydajność.

4.9.2 Komunikacja z czujnikami:

MQTT: Protokół komunikacyjny używany do komunikacji między urządzeniami IoT. Node.js ma wiele bibliotek wspierających MQTT, takich jak mqtt, paho-mqtt czy emqx.

HTTP/REST: Jeśli czujniki mają wbudowane API HTTP, można je bezpośrednio wykorzystać do odczytywania danych.

4.9.3 Przechowywanie danych:

Bazy danych: Możesz użyć baz danych takich jak MongoDB²⁹, PostgreSQL czy MySQL do przechowywania danych z czujników.

NoSQL: Jeśli potrzebujesz szybkiego dostępu do danych, możesz rozważyć użycie baz danych NoSQL, takich jak Redis.

4.9.4 Wizualizacja danych:

D3.js: Biblioteka JavaScript do tworzenia interaktywnych wizualizacji danych.

Chart.js: Prostsza w użyciu biblioteka do tworzenia wykresów.

Przykładowa implementacja:

```
const express = require('express');
const mqtt = require('mqtt');
const app = express();
// Połączenie z brokerem MQTT
const client = mqtt.connect('mqtt://broker.example.com');
// Subskrypcja do tematu z danymi z czujnika
client.subscribe('sensor/data');
// Obsługa żądania GET do pobrania danych z czujnika
app.get('/sensor', (req, res) => {
  client.subscribe('sensor/data', (err) => {
```

²⁸ <https://koajs.com>

²⁹ <https://www.mongodb.com>



```
if (err) {  
  console.error(err);  
  return res.status(500).send('Błąd połączenia z czujnikiem');  
}  
  
// Pobieranie danych z czujnika  
client.subscribe('sensor/data', (err) => {  
  if (err) {  
    console.error(err);  
    return res.status(500).send('Błąd połączenia z czujnikiem');  
  }  
  client.on('message', (topic, message) => {  
    const data = JSON.parse(message.toString());  
    res.json(data);  
  });  
});  
});  
});  
});  
  
// Uruchomienie serwera  
app.listen(3000, () => {  
  console.log('Serwer działa na porcie 3000');  
});
```

W powyższym przykładzie:

- Tworzymy aplikację Express.
- Łączymy się z brokerem MQTT.
- Subskrybujemy temat, z którego otrzymujemy dane z czujnika.
- Obsługujemy żądanie GET do pobrania danych z czujnika.
- Wysyłamy dane z czujnika jako odpowiedź JSON.
- Bezpieczeństwo:
- Używamy HTTPS do zabezpieczenia komunikacji.
- Implementujemy uwierzytelnianie i autoryzację, aby ograniczyć dostęp do danych z czujników.
- Monitorowanie i logowanie:



- Używamy narzędzi takich jak Winston lub Morgan³⁰ do monitorowania i logowania zdarzeń.

Implementacja warstwy WWW z czujnikami przy użyciu Node.js może być skomplikowana, ale daje dużą elastyczność i możliwości. Pamiętaj o odpowiednim zabezpieczeniu i monitorowaniu aplikacji

5 Przetwarzanie danych pojazdów w czasie rzeczywistym przy użyciu Internetu rzeczy (ćwiczenia)

To rozwiązanie tworzy potok przetwarzania danych w czasie rzeczywistym, który służy do pozyskiwania i przetwarzania komunikatów z urządzeń IoT w celu ich analizy na platformie big data w chmurze Azure.

Architektura wykorzystuje następujące usługi Azure:

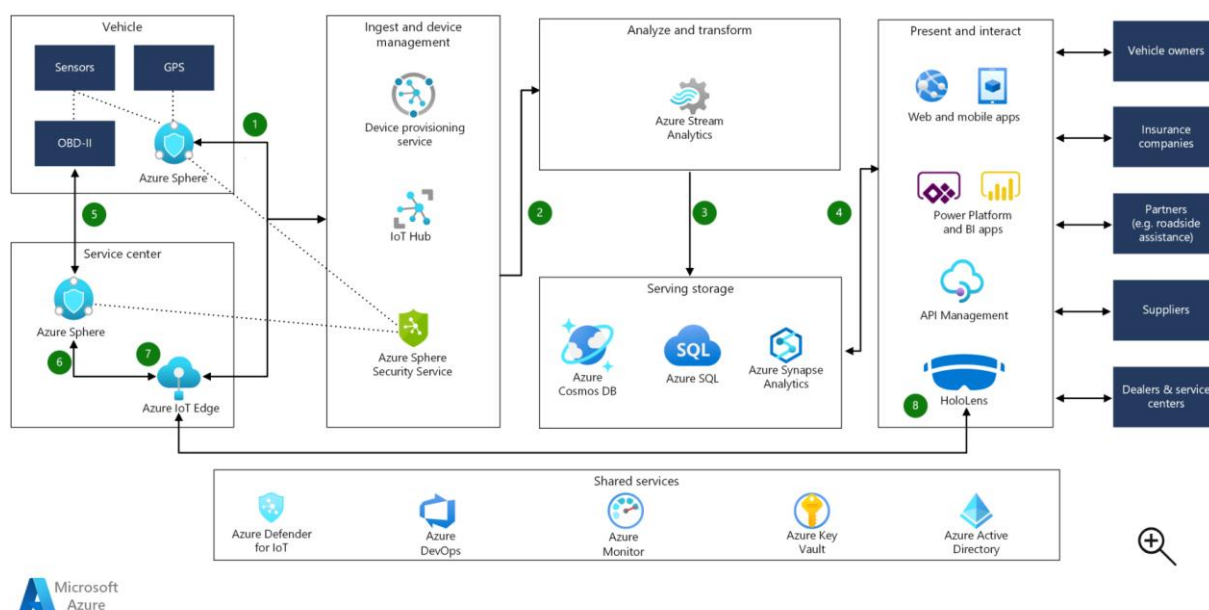
- Azure Sphere: Ta usługa zapewnia bezpieczne i zintegrowane środowisko dla urządzeń IoT, umożliwiając im bezpieczną komunikację i zarządzanie.
- Azure IoT Hub: Jest to centralny punkt zarządzania dla urządzeń IoT, który umożliwia przetwarzanie, routing i analizę komunikatów telematycznych.
- Azure Stream Analytics: Ta usługa przetwarza strumieniowo dane z urządzeń IoT, wykonując na nich różne operacje, takie jak filtrowanie, agregacja i generowanie alertów w czasie rzeczywistym.

Dzięki tej architekturze, komunikaty z urządzeń IoT są zbierane, przetwarzane i analizowane w czasie rzeczywistym, co umożliwia szybkie podejmowanie decyzji i optymalizację procesów. Rozwiązanie to zapewnia skalowalność, bezpieczeństwo i efektywność w przetwarzaniu dużych ilości danych z urządzeń IoT, co jest kluczowe dla współczesnych aplikacji big data.

Na rysunku poniżej pokazano schemat połączeń poszczególnych komponentów.³¹

³⁰ <https://medium.com/@louistrinh/advanced-node-js-api-logging-with-winston-and-morgan-e907a7b15ebc>

³¹ <https://arch-center.azureedge.net/architecture-realtime-analytics-vehicle-data.vsd> plik programu Visio do pobrania



5.1 Przepływ danych

Komunikaty telematyczne³², takie jak prędkość i lokalizacja, są przesyłane z urządzeń wyposażonych w usługę Azure Sphere do usługi Azure IoT Hub za pośrednictwem sieci komórkowej. W scenariuszu greenfield, producent pojazdu może zintegrować moduł Sphere z pojazdem podczas produkcji, natomiast w scenariuszu brownfield, istniejące pojazdy mogą zostać wyposażone w rozwiązanie telematyczne po rynku.

Usługa Azure Stream Analytics pobiera komunikaty w czasie rzeczywistym z usługi Azure IoT Hub, przetwarza je zgodnie z logiką biznesową i przesyła przetworzone dane do warstwy obsługującej, która zarządza różnymi bazami danych. Usługa Azure Cosmos DB przechowuje komunikaty, Azure SQL DB przechowuje dane relacyjne i transakcyjne oraz działa jako źródło danych dla warstwy prezentacji i akcji, natomiast Azure Synapse zawiera zagregowane dane i służy jako źródło danych dla narzędzi analizy biznesowej (BI).

Warstwa obsługująca umożliwia tworzenie aplikacji sieci Web, aplikacji mobilnych, narzędzi analizy biznesowej i rzeczywistości mieszanej. Dane warstwy obsługującej mogą być

³² Słowo "telematyczny" odnosi się do systemów, technologii lub procesów, które wykorzystują telekomunikację do przesyłania danych i informacji na odległość. Jest to połączenie słów "tele" (od telekomunikacja) i "matyczny" (od matematyczny, ponieważ często wiąże się z przetwarzaniem danych). W kontekście motoryzacji, systemy telematyczne odnoszą się do technologii, które umożliwiają zdalne monitorowanie i zarządzanie pojazdami za pomocą sieci komórkowych, satelitarnych lub innych systemów telekomunikacyjnych



udostępniane za pośrednictwem interfejsów API dla firm zewnętrznych, takich jak ubezpieczyciele czy dostawcy.

W przypadku konieczności obsługi pojazdu w centrum dealera, technik serwisowy łączy urządzenie usługi Azure Sphere z portem OBD-II pojazdu. Aplikacja usługi Azure Sphere przesyła strumieniowo dane OBD-II do usługi Azure IoT Edge za pośrednictwem protokołu MQTT, a następnie dane są przesyłane do usługi Azure IoT Hub i przetwarzane w tym samym potoku przetwarzania komunikatów.

W najnowszej wersji systemu operacyjnego 20.10, usługa Azure Sphere może bezpiecznie łączyć się z usługą Azure IoT Edge przy użyciu własnych certyfikatów urządzeń. Certyfikaty są unikatowe dla każdego urządzenia i są automatycznie odnawiane co 24 godziny przez usługę zabezpieczeń Azure Sphere po przejściu przez urządzenie procesu zdalnego zaświadczenia i uwierzytelniania.

Usługa Azure Sphere komunikuje się bezpośrednio z usługą zabezpieczeń Azure Sphere, a nie przez usługę Azure IoT Edge. Usługa zabezpieczeń Azure Sphere to oparta na chmurze usługa firmy Microsoft, która zarządza mikroukładami usługi Azure Sphere, umożliwiając konserwację, aktualizacje i kontrolę.

Brokerowanie MQTT ogólnego przeznaczenia jest dostępne w usłudze Azure IoT Edge. Urządzenie usługi Azure Sphere publikuje komunikaty w wbudowanym temacie MQTT usługi IoT Hub.

Moduły usługi Azure IoT Edge to konteneryzowane aplikacje zarządzane przez usługę IoT Edge, które mogą uruchamiać usługi platformy Azure, niestandardowe modele uczenia maszynowego lub własny kod specyficzny dla rozwiązania.

Technik obsługi, wyposażony w urządzenie HoloLens³³, może subskrybować temat MQTT i bezpiecznie wyświetlać dane OBD-II przy użyciu aplikacji HoloLens zawierającej klienta MQTT. Łącząc urządzenie HoloLens bezpośrednio z bramą usługi IoT Edge, technik może wyświetlać dane pojazdu niemal w czasie rzeczywistym, unikając opóźnień związanych z

³³ <https://www.microsoft.com/pl-pl/hololens> Ergonomiczne, bezprzewodowe, samodzielne urządzenie holograficzne z aplikacjami gotowymi do użycia w przedsiębiorstwie, pozwalające pracować dokładniej i efektywniej.



przesyłaniem danych do chmury i z powrotem. Technik serwisowy może również wchodzić w interakcje z portem OBD-II pojazdu, nawet gdy centrum obsługi jest odłączone od chmury.

5.2 Zastosowanie

Odzyskiwanie, przetwarzanie i wizualizacja danych pojazdu stanowią kluczowe elementy niezbędne do tworzenia zaawansowanych, połączonych rozwiązań samochodowych. Poprzez przechwytywanie i analizowanie tych danych możemy uzyskać cenne, szczegółowe informacje i opracować innowacyjne rozwiązania.

Na przykład w przypadku pojazdów wyposażonych w urządzenia telematyczne, możemy monitorować ich lokalizację w czasie rzeczywistym, planować optymalne trasy, wspierać kierowców oraz wspomagać branże, które korzystają z danych telematycznych, takie jak ubezpieczyciele. Dla producentów pojazdów, informacje diagnostyczne mogą dostarczać istotnych danych dotyczących serwisowania pojazdów i gwarancji.

Dzięki odzyskiwaniu i przetwarzaniu danych pojazdu, możemy tworzyć inteligentne systemy nawigacji, optymalizować zużycie paliwa, przewidywać potencjalne awarie oraz zapewniać lepsze doświadczenie użytkownika. Ponadto, analiza danych telematycznych może pomóc w opracowywaniu nowych usług i produktów, takich jak ubezpieczenia oparte na stylu jazdy czy systemy zarządzania flotą.

6 Wnioski i podsumowanie

Ogólnie rzecz biorąc, wybór platformy zależy od konkretnych potrzeb i wymagań Twojego projektu IoT. Wybór odpowiedniego rozwiązania chmurowego dla aplikacji IoT (Internet of Things) zależy od kilku kluczowych czynników, takich jak wymagania wydajnościowe, skalowalność, bezpieczeństwo, koszty i specyficzne potrzeby projektu. Oto główne cechy platform chmurowych, które mogą być odpowiednie dla systemów IoT:

6.1 Amazon Web Services (AWS)

- IoT Core: Usługa AWS dedykowana dla IoT, która umożliwia zarządzanie urządzeniami, przetwarzanie danych i integrację z innymi usługami AWS.
- Lambda: Funkcje bezserwerowe, które mogą być wyzwalane przez zdarzenia z urządzeń IoT i wykonywać logikę biznesową.
- S3: Magazynowanie danych z urządzeń IoT.

- **DynamoDB:** Baza danych NoSQL do przechowywania danych w czasie rzeczywistym.
- **CloudWatch:** Monitorowanie i logowanie zdarzeń z urządzeń IoT.

6.2 Microsoft Azure

- **Azure IoT Hub:** Usługa dedykowana dla IoT, która umożliwia zarządzanie urządzeniami, przetwarzanie danych i integrację z innymi usługami Azure.
- **Azure Event Grid:** Usługa do wyzwalania funkcji bezserwerowych w odpowiedzi na zdarzenia z urządzeń IoT.
- **Azure Functions:** Funkcje bezserwerowe do wykonywania logiki biznesowej.
- **Azure Blob Storage:** Magazynowanie danych z urządzeń IoT.
- **Azure Stream Analytics:** Przetwarzanie danych w czasie rzeczywistym.

6.3 Google Cloud Platform (GCP)

- **Cloud IoT Core:** Usługa dedykowana dla IoT, która umożliwia zarządzanie urządzeniami, przetwarzanie danych i integrację z innymi usługami GCP.
- **Cloud Functions:** Funkcje bezserwerowe, które mogą być wyzwalane przez zdarzenia z urządzeń IoT.
- **Pub/Sub:** Usługa do przesyłania wiadomości między urządzeniami IoT a backendem.
- **BigQuery:** Baza danych do przechowywania i analizowania danych z urządzeń IoT.
- **Stackdriver:** Monitorowanie i logowanie zdarzeń z urządzeń IoT.

6.4 IBM Cloud

- **IBM Watson IoT:** Usługa dedykowana dla IoT, która umożliwia zarządzanie urządzeniami, przetwarzanie danych i integrację z innymi usługami IBM Cloud.
- **IBM Cloud Functions:** Funkcje bezserwerowe do wykonywania logiki biznesowej.
- **IBM Cloud Object Storage:** Magazynowanie danych z urządzeń IoT.
- **IBM Watson Studio:** Analiza danych i uczenie maszynowe.
- **IBM Cloud Monitoring:** Monitorowanie i logowanie zdarzeń z urządzeń IoT.

6.5 Kryteria wyboru

- **Wydajność i skalowalność:** Upewnij się, że wybrana platforma może obsłużyć przewidywaną liczbę urządzeń i generowanych danych.



- **Bezpieczeństwo:** Sprawdź, jakie mechanizmy bezpieczeństwa oferuje platforma, takie jak szyfrowanie danych, uwierzytelnianie i kontrola dostępu.
- **Koszty:** Porównaj koszty różnych platform, uwzględniając opłaty za użycie i skalowanie.
- **Integracja:** Upewnij się, że platforma może łatwo zintegrować się z istniejącymi systemami i narzędziami.
- **Specyficzne potrzeby:** Rozważ specyficzne wymagania projektu, takie jak analiza danych, uczenie maszynowe czy obsługa określonych protokołów komunikacyjnych.
- **Wybór odpowiedniej platformy chmurowej dla systemu IoT** zależy od konkretnych potrzeb i wymagań projektu. Warto przeprowadzić analizę porównawczą i przetestować różne rozwiązania, aby znaleźć najlepsze dopasowanie.
- Warto również wziąć pod uwagę inne czynniki, takie jak koszty, wsparcie techniczne, integracja z istniejącymi systemami oraz preferencje dotyczące ekosystemu chmurowego. Przed podjęciem decyzji, warto przeprowadzić testy i skorzystać z okresów próbnych oferowanych przez poszczególne platformy.



7 Wykaz literatury

1. Atzori L, Iera A, Morabito G (2010) The Internet of things: a survey. Comput Netw 54:2787–2805
https://www.researchgate.net/publication/222571757_The_Internet_of_Things_A_Survey
2. Miorandi D, Sicari S, De Pellegrini F, Chlamtac I (2012) Internet of things: vision, applications and research challenges. Ad Hoc Netw 10(7):1497–1516
https://www.researchgate.net/publication/235642082_Internet_of_Things_Vision_Applications_and_Research_Challenges
3. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and application. Commun Surv Tutor IEEE 17(4):2347–2376
https://www.researchgate.net/publication/279177017_Internet_of_Things_A_Survey_on_Enabling_Technologies_Protocols_and_Applications
4. Botta A, De Donato W, Persico V, Pescapé A (2016) Integration of cloud computing and Internet of things: a survey. Futur Gener Comput Syst 56:684–700
https://www.researchgate.net/publication/283236612_Integration_of_Cloud_computing_and_Internet_of_Things_A_survey
5. Chase J (2013) The evolution of the Internet of things, Texas Instrum. Dallas, TX, USA, White Paper, p 7 <https://www.ti.com/lit/ml/swrb028/swrb028.pdf>
6. Rani S, Talwar R, Malhotra J, Ahmed SH, Sarkar M, Song H (2015) A novel scheme for an energy efficient Internet of things based on wireless sensor networks. Sensors 15:28603–28626
https://www.researchgate.net/publication/283715763_A_Novel_Scheme_for_an_Energy_Efficient_Internet_of_Things_Based_on_Wireless_Sensor_Networks



7. Sathyamoorthy P, Ngai E (2015) Energy efficiency as a orchestration service for the Internet of things. In: Proceedings of 11th Swedish national computer networking workshop, pp 41–44
https://www.researchgate.net/publication/304407025_Energy_Efficiency_as_an_Orchestration_Service_for_Mobile_Internet_of_Things
8. Evangelos AK, Nikolaos DT, Anthony CB (2011) Integrating RFIDs and smart objects into a unified Internet of things architecture. Adv Internet Things 1:5–12
https://www.researchgate.net/publication/220220623_Integrating_RFIDs_and_Smart_Objects_into_a_Unified_Internet_of_Things_Architecture
9. Juels A (2006) RFID security and privacy: a research survey. IEEE J Sel Areas Commun. 24(2):381–394
https://www.researchgate.net/publication/3236246_RFID_security_and_privacy_A_research_survey
10. Xiaojun C, Xianpeng L, Peng X (2015) IOT-based air pollution monitoring and forecasting system. In: International Conference on Computer and Computational Sciences (ICCCS). IEEE, pp 257–260
https://www.researchgate.net/publication/307777209_IOT-based_air_pollution_monitoring_and_forecasting_system
11. Da Xu L, He W, Li S (2014) Internet of things in industries: a survey. IEEE Trans Ind Inform 10(4):2233–2243
https://www.researchgate.net/publication/270742269_Internet_of_Things_in_Industries_A_Survey
12. Mineraud J, Mazhelis O, Su X, Tarkoma S (2016) A gap analysis of internet-of-things platforms. Comput Commun 89:5–16
https://www.researchgate.net/publication/271855452_A_gap_analysis_of_Internet-of-Things_platforms



13. Wang HI (2014) Constructing the green campus within the internet of things architecture. Int J Distrib Sens Netw 10(3):1–8
https://www.researchgate.net/publication/275594288_Constructing_the_Green_Campus_within_the_Internet_of_Things_Architecture
14. Akkaya K, Guvenc I, Aygun R, Pala N, Kadri A (2015) IoT-based occupancy monitoring techniques for energy-efficient smart buildings. In: Proceedings of IEEE wireless communications and networking conference workshops, pp 58–63
https://www.researchgate.net/publication/280157569_IoT-based_Occupancy_Monitoring_Techniques_for_Energy-Efficient_Smart_Buildings
15. Rani, S., Bhambri, P., Kataria, A., Khang, A., Sivaraman, A.K. “Big Data, Cloud Computing and IoT: Tools and Applications” (1st Ed.) (2023). Chapman and Hall/CRC.
<https://doi.org/10.1201/9781003298335>
16. Perera C, Talagala DS, Liu CH, Estrella JC (2015) Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in IoT clouds. IEEE Trans Comput Soc Syst 2(4):171–181
https://www.researchgate.net/publication/289406827_Energy-Efficient_Location_and_Activity-Aware_On-Demand_Mobile_Distributed_Sensing_Platform_for_Sensing_as_a_Service_in_IoT_Clouds
17. Rohokale VM, Prasad NR, Prasad R (2011) A cooperative Internet of things (IoT) for rural healthcare monitoring and control. In: Proceedings of 2nd wireless communication, vehicular technology, information theory and aerospace and electronic systems technology (Wireless VITAE), pp 1–6
https://www.researchgate.net/publication/224244925_A_cooperative_Internet_of_Things_IoT_for_rural_healthcare_monitoring_and_control



18. Kim J (2015) Energy-efficient dynamic packet downloading for medical IoT platforms. IEEE Trans Ind Inform 11(6):1653–1659 [https://link.springer-com-443.webvpn.synu.edu.cn/chapter/10.1007/978-3-030-82800-4_1](https://link.springer.com/443.webvpn.synu.edu.cn/chapter/10.1007/978-3-030-82800-4_1)
<https://ieeexplore.ieee.org/document/7109892>
19. Jebaraj, L., Khang, A., Vadivelraju, C., Antony, R.P., Kumar, S. (Eds.). “Smart City Concepts, Models, Technologies and Applications”, Smart Cities: IoT Technologies, Big Data Solutions, Cloud Platforms, and Cybersecurity Techniques (1st Ed.) (2023). CRC Press. <https://doi.org/10.1201/9781003376064-1>
20. Khang, A., (2021). Material4Studies, Material of Computer Science, Artificial Intelligence, Data Science, IoT, Blockchain, Cloud, Metaverse, Cybersecurity for Studies. https://www.researchgate.net/publication/370156102_Material4Studies
21. Shaikh FK, Zeadally S (2016) Energy harvesting in wireless sensor networks: a comprehensive review. Renew Sustain Energy Rev 55:1041–1054
https://www.researchgate.net/publication/285782481_Energy_harvesting_in_wireless_sensor_networks_A_comprehensive_review
22. Hubbard B, Wang H, Leasure M (2016) Feasibility study of UAV use for RFID material tracking on construction sites. In: Proceedings of 51st ASC annual international conference proceedings college station, TX, USA
<https://www.semanticscholar.org/paper/Feasibility-Study-of-UAV-use-for-RFID-Material-on-Hubbard-Wang/cbe1e0b894613a13c76bc030062fc59c6a3edc0a>
23. Bhambri, P., Rani, S., Gupta, G., Khang, A. Cloud and Fog Computing Platforms for Internet of Things (2022). CRC Press. <https://www.amazon.com/Computing-Platforms-Internet-Chapman-Society/dp/1032101504>
24. <https://internetofthings.ibmcloud.com>
25. <https://learn.microsoft.com/pl-pl/training/modules/build-web-api-nodejs-express/>



8 Wykaz tabel

Tabela 1 Protokoły stosowane przez platformy chmurowe 24

Tabela 2 Lista właściwości i oczekiwanych rezultatów 48

9 Wykaz rysunków

Rysunek 1 Model chmury czujników [4-7] 10

Rysunek 2 Zastosowanie Internetu rzeczy [6-9] 12

Rysunek 3 wykorzystanie usługi AWS IoT Analytics źródło: <https://aws.amazon.com/iot-analytics> 29

Rysunek 4 Strona logowania AWS 33

Rysunek 5 Przygotowanie kanału 33

Rysunek 6 polecenie tworzenia kanału 35

Rysunek 7 Przyznanie usłudze uprawnień 36

Rysunek 8 Globalna struktura Google Cloud źródło <https://cloud.google.com> 25

Rysunek 9 Architektura z ogólnym produktem platformy IoT działającym w Google Cloud
źródło <https://cloud.google.com/architecture/connected-devices/iot-platform-product-architecture> 26

Rysunek 10 Panel klienta Azure <https://portal.azure.com> 37

Rysunek 11 Pasek manu witryny Azure Portal 37

Rysunek 12 Monitorowanie urządzenia Azure 40

Rysunek 13 Wybieranie okna metryki 41

Rysunek 14 Całkowita liczba wiadomości 42

Rysunek 15 Panel klienta IBM Cloud 44

Rysunek 16 Panel IBM Watson IoT 47

Rysunek 17 Połączenie z Watson IoT 48

Rysunek 18 Hello Word w Node.js 51

Rysunek 19 Instalacja Node.js Windows 54

Rysunek 20 Plik HelloWorld.js 55