PoC (Proof of Concept) – dowód słuszności koncepcji, prototyp rozwiązania.

IoT (Internet of Things) – koncepcja, w której urządzenia mają czujniki, oprogramowanie, łączność z internetem, po to aby mogły: zbierać dane, wymieniać się nimi z innymi systemami, działać autonomicznie.

Podpunkt 1

Cel: wykrywanie anomalii – przegrzania, zatrzymanie cyklu – na maszynach produkcyjnych w czasie rzeczywistym, aby:

- uniknąć awarii, które mogą wiele kosztować

- zminimalizować przestoje w produkcji

- poprawić jakość

Technologie Azure:

|  |  |  |
| --- | --- | --- |
| Warstwa | Narzędzie | Rola |
| Ingest | Azure Iot Hub | Miejsce zbierania danych z czujników na linii produkcji |
| Stream | Azure Stream Analytics | Przetworzenie danych – np. wykrycie przekroczeń |
| Storage | Azure Data Lake Gen2 | Skłądowanie danych surowych, a także przetworzonych |
| AI/ML | Azure Machine Learning | Trening modelu wykrywającego anomalie |
| Alerty | Azure Monitor + Logic Apps | Wysyłanie powiadomienia przy wykryciu anomalii |
| Wizualizacja | Power BI | Dashboard dla inżynierów i operatorów linii produkcyjnej |

Czy warstwy/technologie w PoC zawsze są takie same?

Nie zawsze, ale bardzo często niektóre z nich się powtarzają. Często zmieniają się konkretne narzędzia, ale ogólna struktura pozostaje zbliżona do innych. Nie zawsze musimy używać każdej warstwy. Są przypadki, gdzie użycie każdej warstwy znanej w PoC jest po prostu stratą czasu.

Czy powyższą rozpiskę warstw można skrócić?

Tak, możemy nasze warstwy ograniczyć jedynie do: zebrania danych, treningu modelu do wykrywania anomalii oraz demonstracji wartości

Skrócona wersja miałaby postać:

|  |  |  |
| --- | --- | --- |
| Warstwa | Narzędzie | Rola |
| Ingest | Azure IoT Hub | Zbieranie danych z czujników |
| Analiza + alerty | Azure Stream Analytics | Prosta analiza regułowa + wykrywanie progów |
| ML | Azure ML | Test modelu anomalii |
| Wizualizacja | Power BI | Prosty dashboard do przedstawienia wyników |

Która wersja jest zatem lepsza? (pod kątem interview)

Najpierw można przedstawić krótszą i prostszą wersję, aby pokazać, że rozumiemy architekturę end-to-end, jeśli trzeba to umiemy upraszczać, a następnie ją rozbudować do pełnej wersji produkcyjnej.

Przebieg PoC – dla rozbudowanej wersji

1. Zbieranie danych:

* Czujniki zamontowane na maszynach produkcyjnych
* Komunikacja przez protokół MQTT/http – co to dokładnie znaczy? Oznacza w jaki sposób urządzenia(nadajniki danych) IoT przesyłają dane do chmury (odbiorca danych); protokół – sposób komunikacji między nimi. MQTT jest lżejszy i szybszy – idealny do ciągłego przesyłania danych z produkcji. HTTPS – bardziej ogólny, dobry do prostych zastosowań.
* Dane przesyłane do Azure IoT Hub.

1. Analiza w czasie rzeczywistym:

* Azure Stream Analytics analizuje uzyskane wcześniej dane z czujników:

Np. odnośnie temperatury, wykrywa przekroczenia progów.

* Jeśli zostanie przekroczone – wysyła alert do Azure Monitor/Logic Apps

1. Trenowanie modelu ML

* Dane z Data Lake Gen2 wykorzystujemy do trenowania modelu detekcji anomalii:

Narzędzie – Jupyter Notebooks; model uczy się co to jest „normalne działanie” i wykrywa wszelkie odchylenia

1. Wdrożenie modelu

* Model zostaje wdrożony jako endpoint REST API (adres w internecie, pod który można wysylac żadanie (dane). W projektach IoT/ML/API – to kluczowy sposób komunikacji między systemami).
* Stream Analytics może wysyłać dane do endpointu – predykcja anomalii w czasie rzeczywistym.

1. Wizualizacja i raportowanie

* Power BI pokazuje: dane z ostatnich dni, status czujników, wykresy z oznaczeniem anomalii, aktywne alerty.