Asset Health Monitoring System

Project Overview

Exercise 2: Asset Health Monitoring System

Description: An Asset Health Monitoring System (AHMS) is a comprehensive solution to monitor and manage the condition and performance of physical assets in various industries. This system plays a crucial role in preventive maintenance, ensuring that organizations can identify and address potential issues before they lead to equipment failures or downtime. Implementing an Asset Health Monitoring System can lead to increased operational efficiency, reduced downtime, extended asset lifespan, and improved overall reliability of critical equipment in various industries such as manufacturing, energy, transportation, and more.

Task: Design an architecture for an Asset Health Monitoring System for a manufacturing facility that relies on a range of machinery and equipment. Discuss your approach to designing the system, taking into consideration key components, scalability, security, and future maintenance. Additionally, explain how you would address specific challenges related to real-time health monitoring, data analytics, user interfaces, and compliance with relevant standards.

Technical Implementation

From implementation perspective actor1 is the maintenance person who physically inspect the systems and update the application from there it will be sent to back end microservice and finally stored in OLTP database (Postgres). Application end point will be exposed to end user with Egress router. So that it can be accessed via internet.

UI interface interact with Azure active directory for authentication and authorization and all REST API will be secured with JWT token.

Kafka connect is used to copy the data from OLTP to NO SQL DB in real time. So that it can be accessed by the other microservice for viewing the status on each equipment for respective organizations and other functionality can be accommodated like customization of action or viewing the report.

For any kind of notification separate the microservice is managed, it will take care on sending email basis on user actions or automated email basis on data.

Other side the sensor is installed on each machine will send the data to Kafka topic in JSON format from there Kafka stream will read the data and process them in real time and update the NOSQL DB in real time.

From here Actor 2 will access the data and view the report for each organization and take necessary action and others functionality can be accommodated in this microservice (front end and backend mS) . Here if user can validate the report from actor 1 or form sensor and take necessary action. Basis on action , notification mS will identify and send email to authorized person.

Note:

- In feature if customer ask for data warehousing, We can enhance the existing architecture to extract the data from NOSQL DB and store in Azure BLOB from there Azure data factory can be used to load in to different data base from there we can use Power BI to display the report.
- Application Insights will be used to monitor the microservices performance and logs.

Scalability:

- We will containerize the application using docker and Kubernetes. In docker-compose.yml we will specify the number of nodes needed. Also specify the condition if the CPU & memory consumption of respective pod is more than 90% then other pods will come up automatically. So that we can maintain the scalability of application.
- Implementation of Database indexes (Unique and composite index), writing the optimized query & creation table of partition.
- Implementation of database sharding in required scenarios.
- Implementation of caches for static data retravel.
- Vertical and Horizontal scaling also be utilized.
- Creating the required cloud services in the nearby actual end user location where equipment's are deployed.

Disaster recovery.

- Create the Database replication factor (ZRS, LRS, GRS) so that in case of unknown things happens and it can be accessed from other regions.

Security:

- All microservices secured inside the azure environment and appropriate network IP and port will be enabled to access the application from outside the azure environment.
- Application will be registered with Azure active directory before it will be enabled for end user. So that all authorization and authentication will be handled through active directory.
- API are secured with JWT token on top of active directory.

Health monitoring

- Azure Application Insights: This service is designed to monitor the performance and usage of your applications. It helps you detect and diagnose issues and understand user behavior. Key features include:

  1) Application Insights can be integrated with your microservices to monitor their performance and detect issues.
  2) It supports distributed tracing, which allows you to track requests as they flow through different microservices.
  3) You can monitor dependencies such as HTTP calls, SQL databases, and other services your microservices interact with.
  4) Application Insights provides rich analytics and diagnostics tools to help you diagnose performance bottlenecks and failures.
  5) Customized way is create the separate the microservice to and create a method that will be triggered to actuator end point for every minutes and check for status.

Assumption:

- Sensor installed in each machine will be send JSON data about the current status or any issues.
- Archive of data is not addressed as part of this implementation.

Technical Stack used in project.

Software Used	Version
Spring boot	3.0.X
Java	17
Azure Kubernetes Service	1.29
Docker	24.0.
Application Insight	
Azure container service	
Azure active directory	
Postgres	13
React	
Mango DB	4.0
Kafka Stream	
Kafka Connect	