Group Project Report

Real-Time Patient Vital Monitoring System

Improving Remote Patient Monitoring Systems

By:

Shriansh Nauriyal & Swaroop Malina

U35276962 & U87601026

CS777 A1 Big Data Analytics (Fall 2023)

Index

Problem Statement

In an era dominated by technological advancements and a burgeoning digital healthcare landscape, the need for an efficient and proactive Real-time Patient Monitoring System has become increasingly evident. The project aims to address the growing demand for continuous, remote monitoring of crucial vital signs, ensuring timely intervention and improved patient outcomes.

The Global Internet of Medical Things (IoMT) market is projected to soar to $187.6 billion by 2028, more than four times its worth in 2020 according to Fortune Business Insights, signifying a paradigm shift towards interconnected healthcare solutions. With over 70 million users expected to embrace Remote Patient Monitoring (RPM) in the United States by 2025, the urgency to enhance healthcare delivery through innovative technologies is paramount.

Even in the present day, smart wearables like the Apple Watch have set the stage by delving into the measurement of vital metrics such as SPo2 and blood pressure. These early forays into wearable health tech underscore the potential for seamlessly integrating technology into patient care, creating a precedent for more sophisticated and dedicated solutions.

Despite the strides made in remote monitoring technology, there exists a critical gap in the real-time monitoring of patients' vital signs by health professionals, and more importantly generating instantaneous alerts which are communicated to doctors and other medical support staff automatically, necessitating a dedicated and comprehensive solution. The Real-time Patient Monitoring System seeks to bridge this gap by continuously tracking essential metrics including blood pressure, heart rate, SPo2, respiration rate, and temperature and communicating any outliers to medical professionals as soon as possible.

The overarching problem is twofold: ensuring the reliability and accuracy of real-time data acquisition, and developing an intelligent alert mechanism that promptly notifies healthcare providers when patients' vital signs deviate from established norms. This necessitates a robust architecture that can handle the influx of data, analyze it in real-time, and initiate automated alerts via email or SMS, facilitating swift and informed medical interventions.

Features

The Real-time Patient Monitoring System boasts a robust set of features designed to seamlessly integrate with cutting-edge technologies, ensuring proactive and efficient healthcare interventions. The features of the system are as follows:

1. The system **supports real-time data streams from multiple patient remote vital sensing IoMT devices**. Leveraging Kafka Streams, it efficiently captures records from patients every 30 seconds, encompassing vital information such as blood pressure, heart rate, SPo2, respiration rate, and temperature.

2. To ensure the utmost patient safety, the system employs Spark Streaming to **analyze vital signs against predefined threshold values**. When any vital surpasses its safe limit, the Spark application promptly **generates alert messages**. These alerts, along with relevant patient information, are then seamlessly transmitted to AWS SNS (Simple Notification Service).

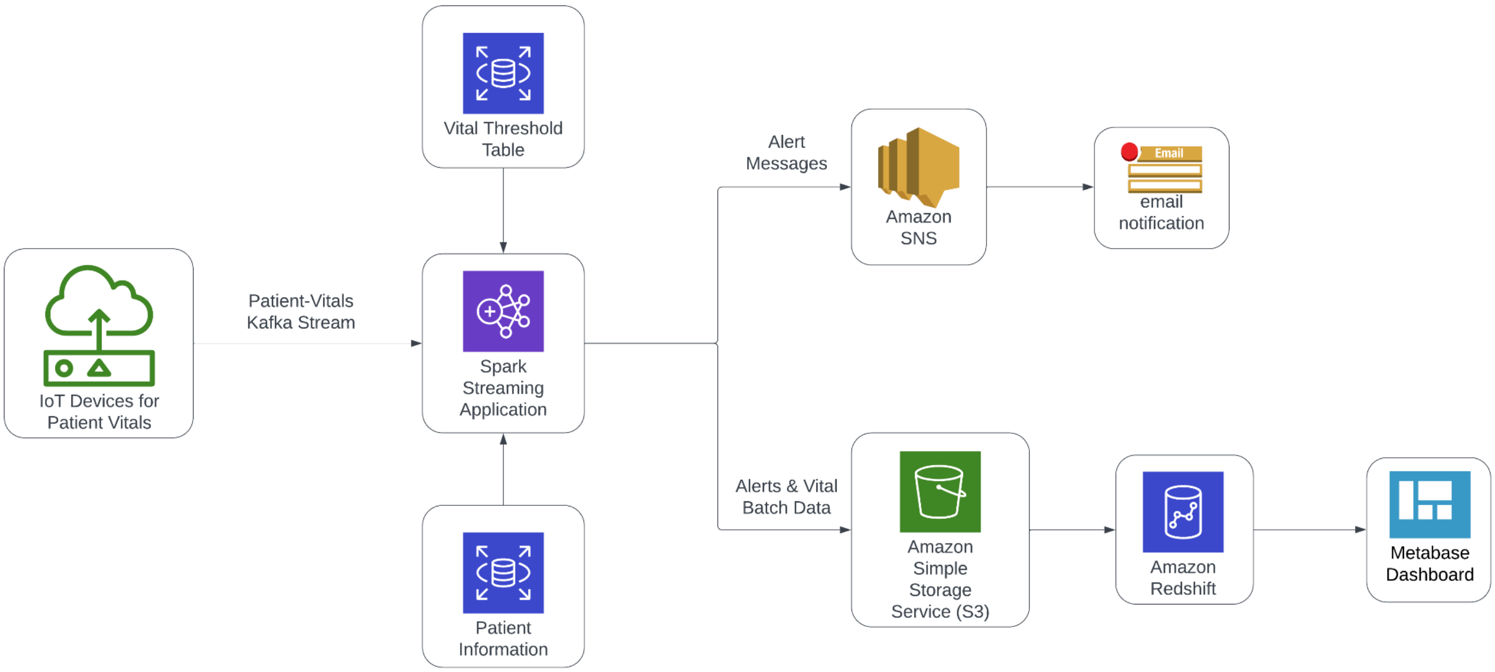
3. AWS SNS serves as the pivotal bridge, **taking charge of disseminating alerts and patient data to designated healthcare professionals via email**. This ensures that timely action can be taken based on real-time patient vitals.

4. In parallel, the Spark application enriches the data landscape by **sending both the patient vitals stream and processed alert data to AWS S3**. This strategic move facilitates future analysis, offering a treasure trove of information for gaining deeper insights into patient trends and system performance.

5. The analytical prowess of the system extends further with Metabase and AWS Redshift integration. By connecting seamlessly to AWS S3, AWS Redshift creates the tables on the Redshift server and then Metabase unleashes its capabilities in **generating insightful data visualizations**. This feature empowers medical professionals with a comprehensive and visually intuitive understanding of patient data trends, facilitating informed decision-making.

Together, these features constitute a powerful ecosystem aimed at revolutionizing patient care and contributing to the evolution of modern healthcare practices.

Project Workflow



The flow of the entire project setup is as follows:

- The project starts with patient vital data coming from IoT devices via the Patient-Vitals Kafka Stream. This stream has data across multiple patients and it gets new data every 30 seconds.

- Apart from the Kafka stream, we also have two batch data sources - Vital Threshold data and Patient Information data. The Vital threshold data contains the upper and lower safe limits of the patient vital metrics. The Patient information table has personal data related to the Patients such as their name, address, phone number, admitted ward number and so on.

- This data is important since it will be included in the Alert messages that are sent to the medical professionals since patients could also have been discharged and in case of certain emergencies, they might have to directly go to the patient's address and/or contact their phone number.

- There is a central Spark Streaming application which handles these three data sources. The Spark Streaming application is responsible for two main tasks:

- Firstly, it has to compare the patient's vital stream against the threshold data to check if any of the vitals are crossing the safe threshold values. If that does happen, then it generates an appropriate alert message to be sent to AWS SNS.
- Secondly, the application also exports the patient vital data as well as the processed alerts data to an S3 bucket where it can then be used later for Data analysis and visualization with the help of AWS Redshift and Metabase.

- Amazon SNS is then responsible for receiving the alert data stream and then it sends these alerts to the relevant medical professionals via emails. Please note that the system is also able to send these messages via SMS

- AWS Redshift is used here as one of the tools for performing SQL like queries to create the schema and load the data from S3 to create the three tables.

- Metabase is then being used here where it is utilizing the Redshift tables for creating a dashboard to showcase some important data visualizations.

Redshift Tables

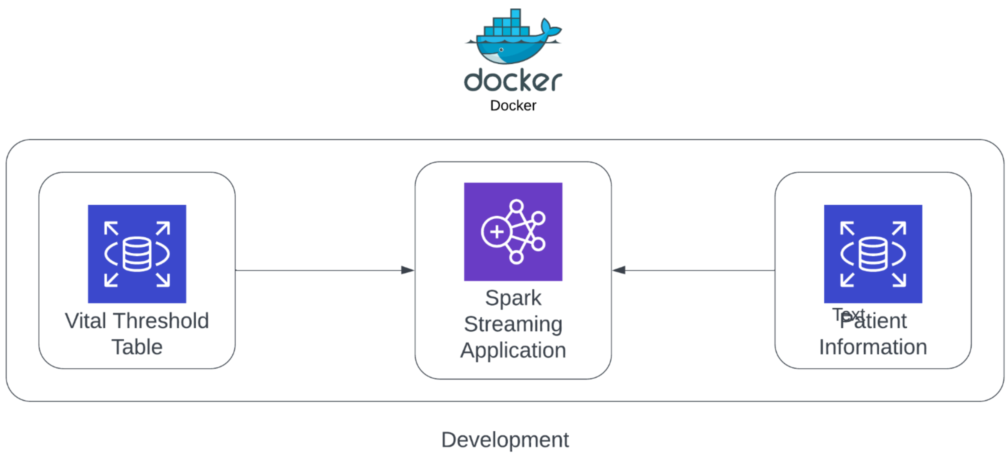
Below is the schema of the Redshift tables that are being used in the Data Visualization part. The schema is as expected. This data will be loaded to Redshift and Metabase will then connect to these tables for creating the data visualizations:

patient_vitals	
patient_id	INT
heart_rate	INT
systolic_bp	INT
diastolic_bp	INT
temperature	FLOAT
respiration_rate	INT
spo2	FLOAT
date_time	TIMESTAMP

alerts	
patient_id	INT
alert_metric	VARCHAR(255)
value	FLOAT
threshold_range	VARCHAR(255)
alert_timestamp	TIMESTAMP

patient_information	
patient_id	INT
patient_name	VARCHAR(255)
phone_number	VARCHAR(255)
age	INT
admitted_ward	INT
address	VARCHAR(255)

Docker Development Environment



Docker is a platform for developing, shipping, and running applications in a consistent and portable environment. It utilizes containerization technology, enabling developers to package an application and its dependencies into a lightweight, self-sufficient unit called a container. These containers can run on any system that supports Docker, ensuring consistent behavior across different environments. Docker simplifies the deployment process, enhances scalability, and facilitates collaboration by isolating applications from underlying infrastructure, streamlining development workflows, and promoting a modular approach to software architecture.

In our project, Docker has played a pivotal role in streamlining our development environment. Leveraging Docker and Docker Compose, we deployed essential services such as Apache Spark, Kafka, and Zookeeper. Docker Compose proved instrumental in seamlessly starting these services, ensuring a consistent and reproducible development environment across different setups. The utilization of Dockerfiles facilitated the creation of a unified container housing all the requisite dependencies, simplifying the deployment process.

Within this Dockerized ecosystem, we established a network where two containers could efficiently communicate with each other. This approach not only enhanced the efficiency of our development workflow but also ensured the seamless integration and interaction of various components within the Dockerized environment. The encapsulation of services in Docker containers significantly contributed to the ease of management and reproducibility of our development environment.

Big Data Technologies

The Real-time Patient Monitoring System is underpinned by a sophisticated technological framework, seamlessly integrating various tools to optimize healthcare processes.

1. **Apache Kafka**:

The system leverages Apache Kafka to facilitate the Patient-Vitals Kafka Stream. This open-source stream processing platform ensures the seamless flow of real-time patient vital data from IoT devices. With a publish-subscribe model, Kafka enables scalable and fault-tolerant data streaming, ensuring a reliable influx of data every 30 seconds.

2. **Apache Spark Streaming:**

Spark Streaming, as part of the central Spark application, plays a dual role—comparing patient vital streams against predefined threshold values and exporting processed data to AWS Redshift for future analysis. It ensures that the system remains agile and responsive, laying the foundation for real-time alerting and comprehensive insights into patient health dynamics.

3. **Amazon S3**:

Amazon S3 serves as the central repository for batch data sources, including Vital Threshold Data and Patient Information Data. Storing data in S3 provides a scalable and durable solution, allowing for efficient data retrieval and export. This setup enhances accessibility and supports the system's data analysis needs.

4. **AWS Redshift**:

As a robust data warehouse, AWS Redshift plays a critical role in data analysis. Connected directly to the S3 bucket, it performs SQL-like queries on the exported patient vital data and processed alerts to create their corresponding tables.

5. **AWS SNS (Simple Notification Service)**:

AWS SNS is employed to handle the alerting process. It seamlessly receives the alert data stream generated by the system and dispatches alerts to relevant medical professionals via email. The flexibility to send messages via SMS enhances communication accessibility, ensuring timely response to critical healthcare situations.

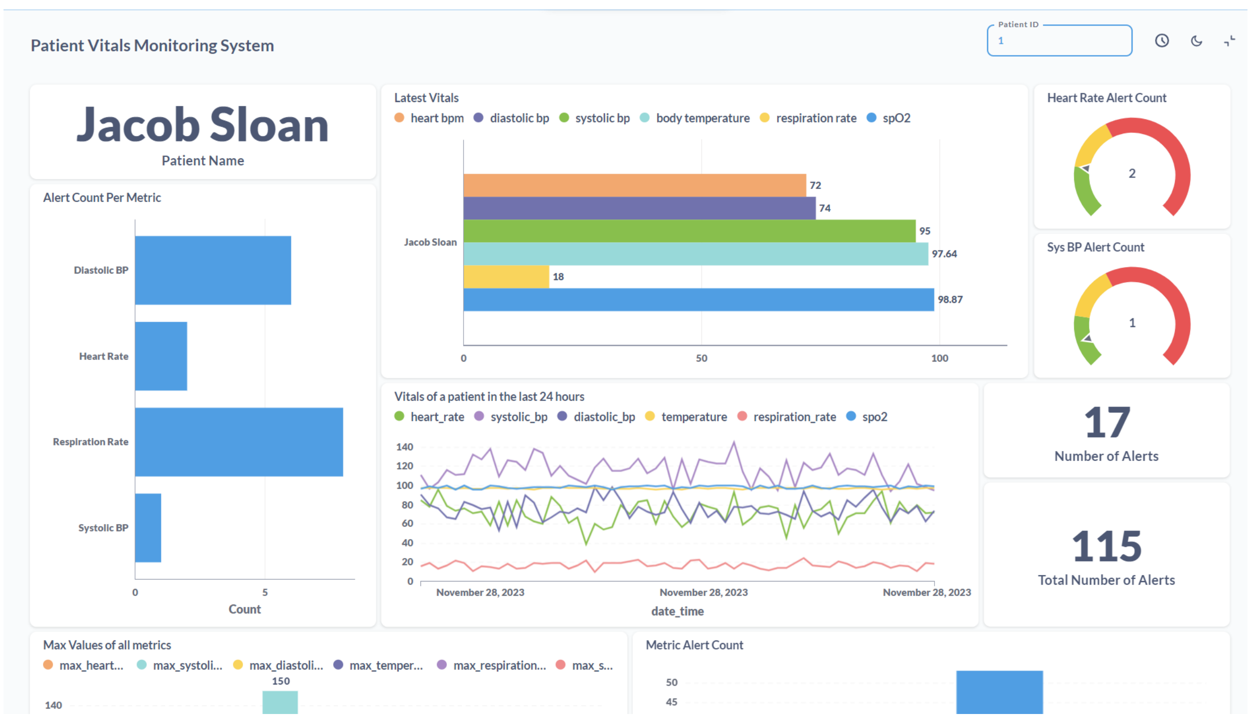
6. **Metabase**:

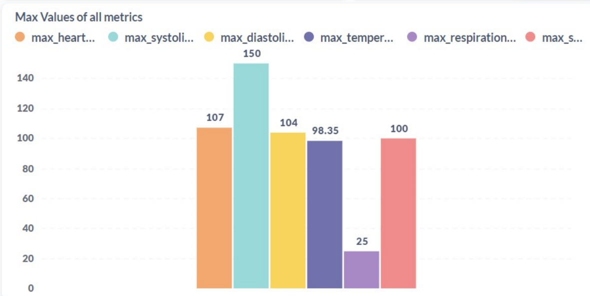
Functioning as the system's key data visualization tool, Metabase plays a crucial role in translating raw patient data into dynamic visualizations. Connected directly to the Redshift server, Metabase extracts valuable insights, offering healthcare professionals an intuitive dashboard to explore and interpret patient vitals and alert data.

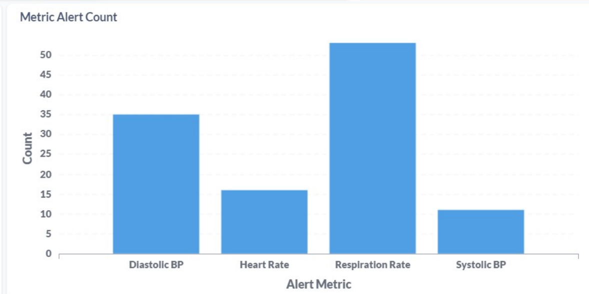
Results

In this project, we successfully implemented a real-time patient monitoring system which monitors patient vitals data, including heart rate, blood pressure, temperature, respiration rate, and SpO2, were simulated using Python scripts and streamed to a Kafka topic. A Spark Structured Streaming application consumed this data, and alerts were triggered when vital signs crossed predefined threshold values. AWS Simple Notification Service (SNS) was employed to notify doctors of critical patient conditions promptly. Additionally, a comprehensive analytics dashboard was established using Metabase, connected to Redshift tables housing the patient vitals data. The dashboard provides a user-friendly interface for healthcare professionals to monitor patient health trends, analyze historical data, and make informed decisions. The successful integration of streaming analytics, alerting mechanisms, and a powerful analytics dashboard enhances healthcare professionals' ability to respond proactively to critical patient conditions, thereby improving overall patient care.

Metabase Dashboard:







Latest Alerts

patient_id	patient_name	alert_metric	value	threshold_range	alert_timestamp
9	Gregory Gallegos	Respiration Rate	21	(8-20)	November 28, 2023, 7:45 PM
4	Stefanie Turner	Diastolic BP	98	(50-90)	November 28, 2023, 7:45 PM
9	Gregory Gallegos	Heart Rate	46	(50-100)	November 28, 2023, 7:43 PM
9	Gregory Gallegos	Systolic BP	143	(50-140)	November 28, 2023, 7:42 PM
5	Andrew Jones	Respiration Rate	21	(8-20)	November 28, 2023, 7:42 PM
4	Stefanie Turner	Respiration Rate	21	(8-20)	November 28, 2023, 7:42 PM
8	Mr. William Mccoy	Diastolic BP	91	(50-90)	November 28, 2023, 7:42 PM
5	Andrew Jones	Diastolic BP	93	(50-90)	November 28, 2023, 7:41 PM
3	Matthew Williams	Respiration Rate	21	(8-20)	November 28, 2023, 7:41 PM
1	Jacob Sloan	Diastolic BP	96	(50-90)	November 28, 2023, 7:41 PM

Notification Alert:

1 of 9

AWS Notification Message Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com>
to me

2:36 PM (9 hours ago)

Alert:
Patient: Gregory Gallegos,
Address: 73 Maple Street, 02136,
Respiration Rate is abnormal.
Current value: 21.00
Must be in: (8, 20)

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:903187628715:patient_vital_alert:8d0fc070-8889-43c9-b76a-cd8dfaaf9889&Endpoint=patientvitalmonitoringsystem@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Conclusion

In conclusion, the Real-time Patient Monitoring System presents a cloud-powered solution at the intersection of healthcare and technology. By harnessing the power of Apache Kafka, Amazon S3, AWS Redshift, AWS SNS, and Metabase, the system establishes a robust framework for continuous patient vital monitoring, alert generation, and insightful data visualizations.

The integration of Apache Kafka ensures a reliable and real-time flow of patient data, setting the stage for instantaneous insights. Storing batch data in Amazon S3 not only enhances accessibility but also provides a scalable solution for efficient data retrieval and export.

AWS Redshift emerges as a pivotal tool for data visualization and storage, seamlessly querying data directly from the S3 bucket to create the required tables. This facilitates the generation of valuable insights, empowering healthcare professionals to make informed, data-driven decisions.

The alerting mechanism, orchestrated through AWS SNS, ensures swift communication with medical professionals via email and SMS. This adaptability enhances the system's responsiveness to critical healthcare scenarios.

Metabase, connected to the S3 bucket, brings the data to life through dynamic and interactive visualizations in the form of a dashboard. This dashboard not only provides a comprehensive overview of patient vitals but also serves as a tool for healthcare professionals to derive meaningful insights.

Looking ahead, the Real-time Patient Monitoring System holds promising avenues for future enhancements. One notable avenue involves the incorporation of custom vital metrics tailored to specific patient conditions or medical scenarios. By expanding the range of monitored parameters beyond conventional vitals, the system could offer a more nuanced and personalized approach to patient care. Additionally, integrating machine learning algorithms could further refine the threshold values, adapting them dynamically based on individual patient trends. The potential inclusion of wearable devices with advanced sensors and continuous monitoring capabilities could also broaden the scope of real-time data collection, enhancing the system's overall effectiveness. These future improvements not only align with the ever-evolving landscape of healthcare technology but also position the project to be at the forefront of innovation in remote patient monitoring.

References

- Hassan, F., E., M. and Sahal, R. (2020). Real-Time Healthcare Monitoring System using Online Machine Learning and Spark Streaming. *International Journal of Advanced Computer Science and Applications*, 11(9). doi:https://doi.org/10.14569/ijacsa.2020.0110977.

- Meola, A. (2022). *IoT Healthcare in 2022: Companies, medical devices, and use cases*. [online] Insider Intelligence. Available at: https://www.insiderintelligence.com/insights/iot-healthcare/ [Accessed 28 Nov. 2023].

- Meola, A. (2022). *IoT Healthcare in 2022: Companies, medical devices, and use cases*. [online] Insider Intelligence. Available at: https://www.insiderintelligence.com/insights/iot-healthcare/ [Accessed 28 Nov. 2023].