IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HEALTHCARE AUTOMATED MONITORING SYSTEM

Mrs .J.Princess Bala,M.E.,P.Shamile,A.Thamarai Pushpam,J.Usha,R.Vinisha

Supervisor[1],Student[2]

Computer Science and Engineering

Jayaraj Annapackiam CSI College of Engineering, Nazareth, India

Abstract- Amazon web service(AWS) be building up on the IoT Core and DynamoDB lab. The device simulator will publish HeartRate, SpO2(Oxygen Saturation) and Body Temperature that to the cloud (Iot Core →DynamoDB),and then aggregate data and detect a nomalies. The initial part of the project IoT Core lab. The device IOT will publish heart rate, oxygen saturation and body temperature values periodically. The device simulator code can virtually developed with the aws and can to push data to an iot core endpoint. We insert values in python code as sensor values and publishing in IOT core. After publishing we are processing in dynamo db to analyze the values like raw data, aggregate data and anomalies data. This project is Similar to the lab, create Things, Certificates, Policy , DynamoDB table (bsm_data) and IoT Core Rule to push the data created to the DynamoDB table. The simulator code is similar to the lab's simulator.Should create at least devices/things. Can modify the simulator code to take device id as another parameter or run two copies in parallel with different hard-coded device ids. Please run them for at least an hour continuously to generate enough data. The data is being generated at 1, 10, and 15 second frequencies for the three different sensor types.

***Index terms*–Health Care, AWS, Monitor, Body Temperature**

I. INTRODUCTION

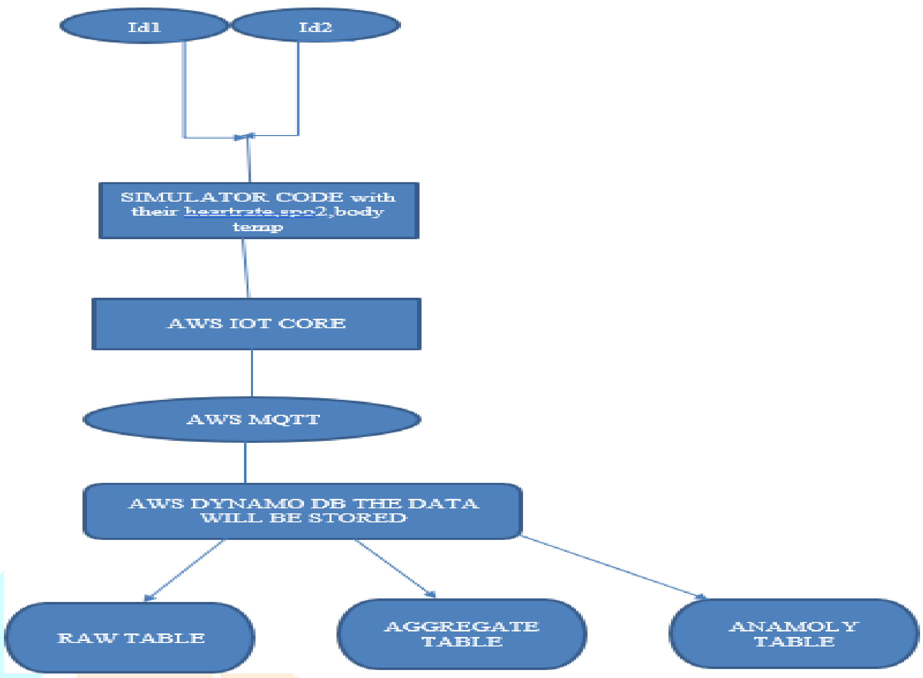
Create Things, Certificates, Policies, Dynamo DB tables (bsm_ data), and IoT Core Rules to transfer the created data to the Dynamo DB tables in a manner similar to the lab. The simulator code is comparable to that used in the lab. Should design at least two objects. The simulator code can be changed to accept the device id as an additional argument or can run two copies simultaneously with various hard-coded device ids. To generate adequate data, kindly run them for at least 1 hour constantly. For the three separate minutes taken into consideration, the data is created at intervals of 1, 10, and 15 seconds. The detain another ratable a because we are merely imitating an Aggregate time frame and immediately issue an alarm. These notifications should be produced on the console, saved to the bsm_alerts table with an unsuitable format, and include the device id, datatype, time of the first incident of breach, the rule that was violated (Hard), and the first instance of breach. Use rules to find anomalies and store the You can use any format, including json, to create a rules config file. It ought to be able to set up rules for combination alerts. For any of the three data kinds, it should be possible to define an OR rule with min and max average values. A trigger count should also be able to be defined in order for the values to be breached continually at that number. Provide functionality to aggregate the data from the three sensors individually, at the minute level should calculate and store the average, minimum and maximum values taking all values with in a minute in account. Since we are only mimicking a continuous monitoring system please provide the aggregator method call to be for a specific time range and aggregate data per minute within that range. This should be stored programmatically in the aggregator table. For example, when called with a time range of '2021-02-17 13:00:00 to 2021-02-17 14:00:00', it should create 60 rows per sensor type in the aggregated table, each being the aggregation of all values with in a minute for that sensor datatype. Entries can be skipped if there is no relevant data Detect anomalies base do n rules and store them a. Create a rules config file using any format that see fit, including j-son . This should have the ability to define combination alert rules. One should be able to define an OR rule with min and max average values for any of the three data types. In addition, one should be able to define a trigger count so that the values should be at least two rules specified in the config file for demonstration.

II.LITERATUREREVIEW

Healthcare is a vast arena that is composed of many different components. Delivering healthcare involves clinical practices, hospitals, pharmacies, home health providers, long term care providers, pharmaceutical companies, and medical-device manufacturers. It also involves health and wellness products and services, insurance companies, and governments providing services to end-users . This section provides a review with an analysis of the recent research on IoT-based healthcare-monitoring systems. summarizes some of the recent studies regarding IoT-based healthcare-monitoring systems. The wearable device developed by Wu et al. monitors various physiological parameters, including body temperature (BT), electrocardiograph (ECG), and heart rate (HR). Using Pulse Arrival Time (PAT) to measure ECG and PPG, it is possible to estimate blood pressure (BP). The interaction between humans and remote monitoring programs is straightforward because all the components are designed within a rigid framework. In addition, the power consumption of the devices is low, and they can communicate wireless to make tailored measurements of a specific physiological signal. The physiological measurements can be wireless transmitted to a gateway using a BLE module. The data are encrypted at the sensor patch and gateways to maintain privacy, ensuring transmission security. The wearable sensor system is connected to the cloud
using a smartphone and a Raspberry Pi module as a gateway; the data can be retrieved and analyzed from the cloud. low energy consumption, BLE technology is unsuitable for wireless communication over long distances and high data rates. Islam et al. developed an intelligent monitoring system for use in a hospital. It not only collects data on patients' BT, HR, and other vital signs but also monitors environmental factors in the hospital room, such as CO, CO2, and humidity. The success rate of modern healthcare systems is ~95% agreement between monitored and actual data in all cases. Medical staff can view the data in real-time, either on-site or remotely. Hypothetically, the technology would be helpful during medical crises and epidemics, as medical personnel would have almost instant access to raw data. The prototype created is incredibly easy to design and use. Such devices could be helpful in managing infectious-disease outbreaks, such as COVID-19. Potentially, this system could save more lives by improving the efficiency of the existing healthcare system. However, at this stage, the system still lacks some epidemic-related sensors that need to be evaluated once implemented. In , Al-Sheik and Ameen propose an IoT health-monitoring system for cell phones that remotely monitors patients' vital signs, including BT, ECG, and blood-oxygen saturation (SpO2). Arduino was used to measure and process this system. This system uses Wi-Fi to send the data to a cloud service on the IoT platform called Blynk; the data can be monitored in real-time. For security and privacy reasons, the results are sent to a specific smartphone that the doctor can monitor. Therefore, two micro controllers, Arduino and Node MCU, are used, which still need to be improved. For long-distance transmission, Wi-Fi technology is not the ideal option. Hamim et al. present an IoT-based health care monitoring system for patients and older adults based on an Android application. The sensors in this prototype collect BT, HR, and Galvanic Skin Response (GSR) data that are fed into a single system, the Arduino Uno platform. Raspberry Pi transfers the data to cloud storage. Android Studio was used to develop the Android app, in which health parameters collected from patients can be visualized. Doctors can use the application to prescribe necessary prescriptions and track the patient's health over time. Using Raspberry Pi 3, Swaroop et al. developed an IoT-based real-time health monitoring system. Data creation, acquisition, processing, communication, and access are the main phases of the system structure. Health data such as HR, BT, and BP were measured. The data are transmitted through modes such as BLE, GSM, and Wi-Fi, i.e., mobile applications, messaging services, and the Internet. It was found that the latency is low, and there is no significant delay between sending and receiving data. Thus, the system's accuracy is limited to the accuracy of the sensors. Gupta outlines a healthcare-monitoring system using the IoT for obese patients. The prototype is a fully functional device that measures body characteristics such as HR, SpO2, BP, and BT. This device is ideal for regular monitoring of body conditions.

III.RESEARCH METHODOLOGY

We build our AWS storage into dynamo DB



AWS(AMAZONWEBSERVICE)

Amazon Web Service is an online platform that provides scalable and cost-effective cloud computing solutions. AWS is a broadly adopted cloud platform that offers several on-demand operations like compute power, database storage, content delivery, etc. ,to help corporate scale and grow.

CLOUD COMPUTING

Cloud Computing is the delivery of services (such as servers, databases, software) tousers. With the help of cloud computing, storing data on local machines is not required. It helps access data from a remote server. Moreover, it is also used to store and access data from anywhere across the world.

MQTT

MQTT represents Message Queuing Telemetry Transport. It is a highly light weight and publish-subscribe messaging transport protocol. This protocol is useful for the connection with there mote area where the bandwidth is excellent. This nature makes it beneficial in multiple situations, including a stable environment such as communication device to device and the internet

IOT (Internet OF Things)

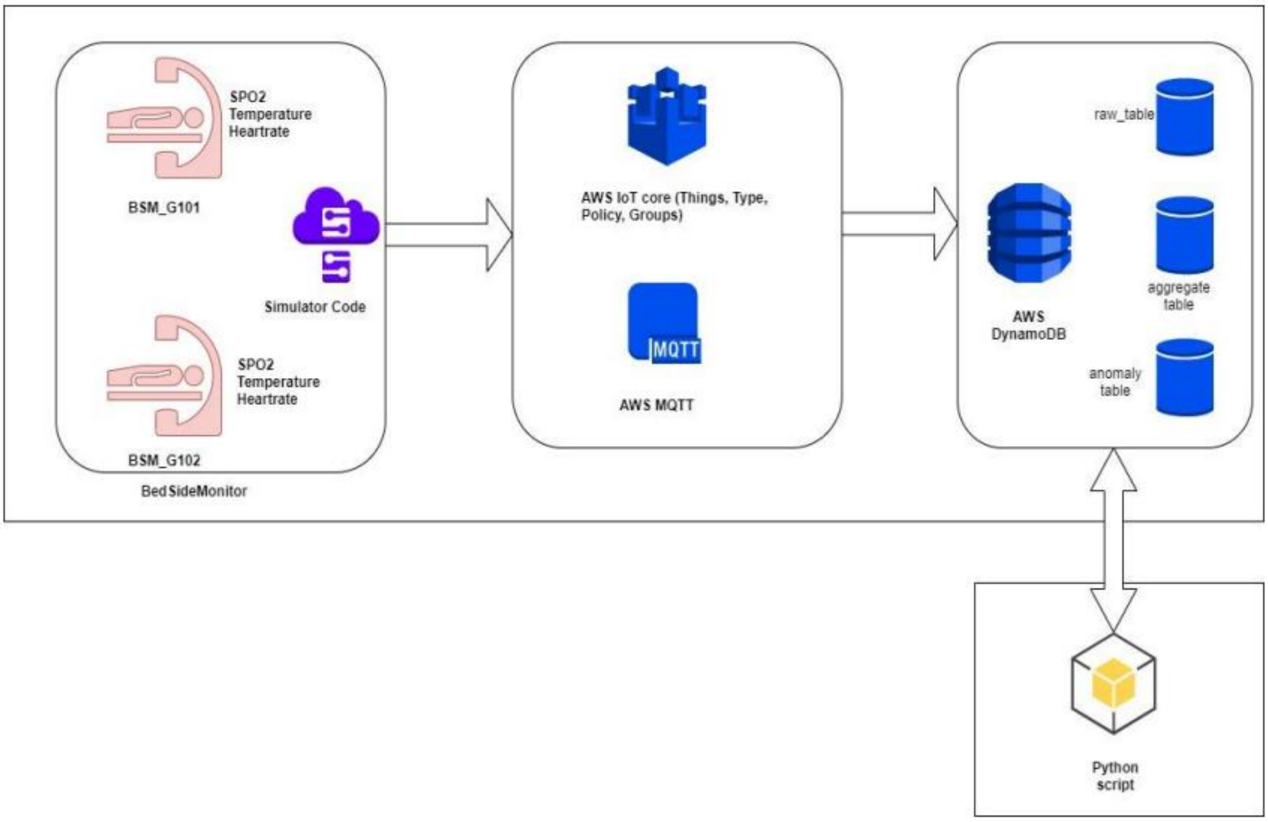
AWS IoT provides the cloud services that connect IoT devices to other devices and AWS cloud services. AWS IoT provides devices software that can help integrate all the IoT devices into AWS IoT-based solutions. If devices can connect to AWS IoT, AWS IoT can connect them to the cloud services that AWS provides.

DYNAMODB

Amazon DynamoDB is a fully managed NoSQL database service that provide fast and predictable performance with seamless scalability With DynamoDB can create database tables that can store and retrieve any amount of data and serve any level of request traffic.

.

IV.SYSTEMARCHITECTURE



- This is a simple example and may not follow some standard practices.
- The focus is on the main flow, and not as much on error handling.
- There will be some differences from standard flow since we are mimicking 'continuous monitoring by an always-on service' with a manually triggered program BedSideMonitor.py - This is quite similar to the device simulator in IoT Core and DynamoDB lab.

(Hard) Detect anomalies

a.Create a rules config file using any format that see fit, including json. This should have the ability to define combination alert rules. One should be able to define an OR rule with min and max average values for any of the three data types. In addition, one should be able to define a trigger count so that the values should be breached continuously that many times, before an alert should be raised. For example, a rule can be {'type': 'HeartRate', 'avg_min': 55, 'avg_max': 105, 'trigger_count': 5} This should trigger an alert if five continuous aggregated (at the minute level) average values are outside the min/max range. Please write an appropriate parser for these rules in the code. There should be at least two rules specified in the config file for demonstration purposes.

b. Create a new table (bsm_alerts) using code or manually via the UI. Create the appropriate model to access it. Provide functionality to run the rule check for a specified time range. It should check all rules on all devices individually within that time period and raise alerts as appropriate. These alerts should be printed on the console, and saved to the bsm_alerts table with an appropriate format, specifying device id, data type, the timestamp of the first instance of breach, and the rule that was breached.

V.MODULESDESCRIPTION

ADMIN MODULE

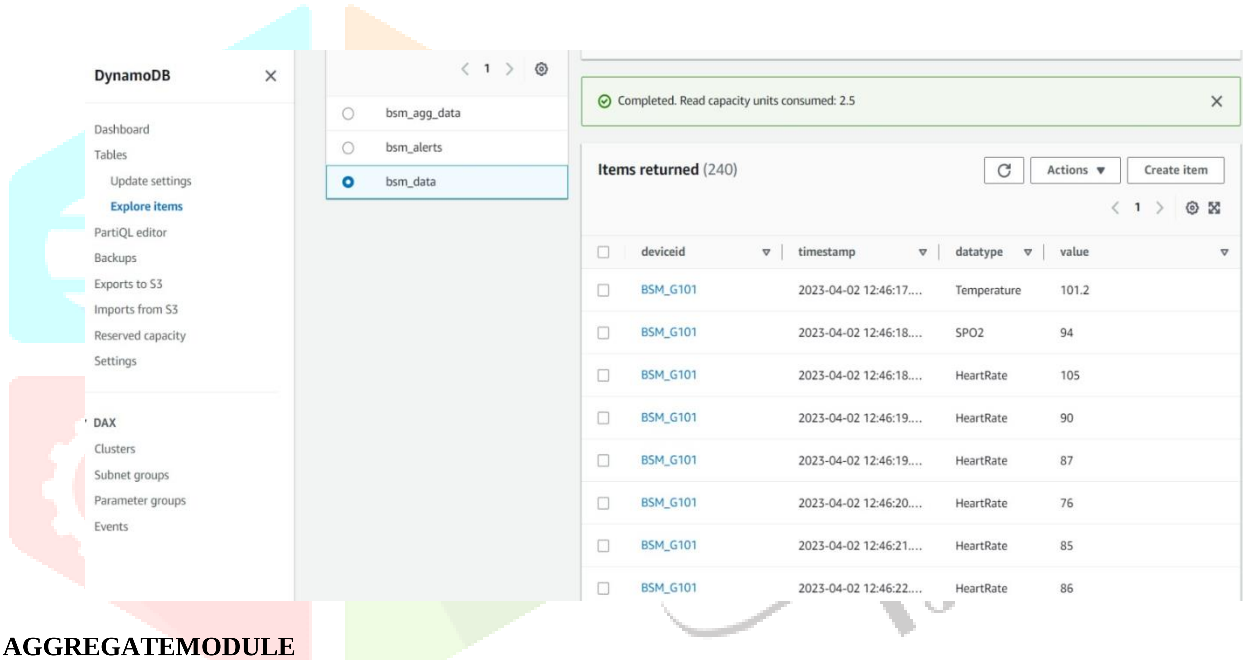
Doctor can monitor patient health condition such as body temperature, oxygen saturation, heartbeat rate. And it is used when the patient is in critical condition.

USER MODULE

Patient can know their health condition, and it is easy to monitor their health condition properly

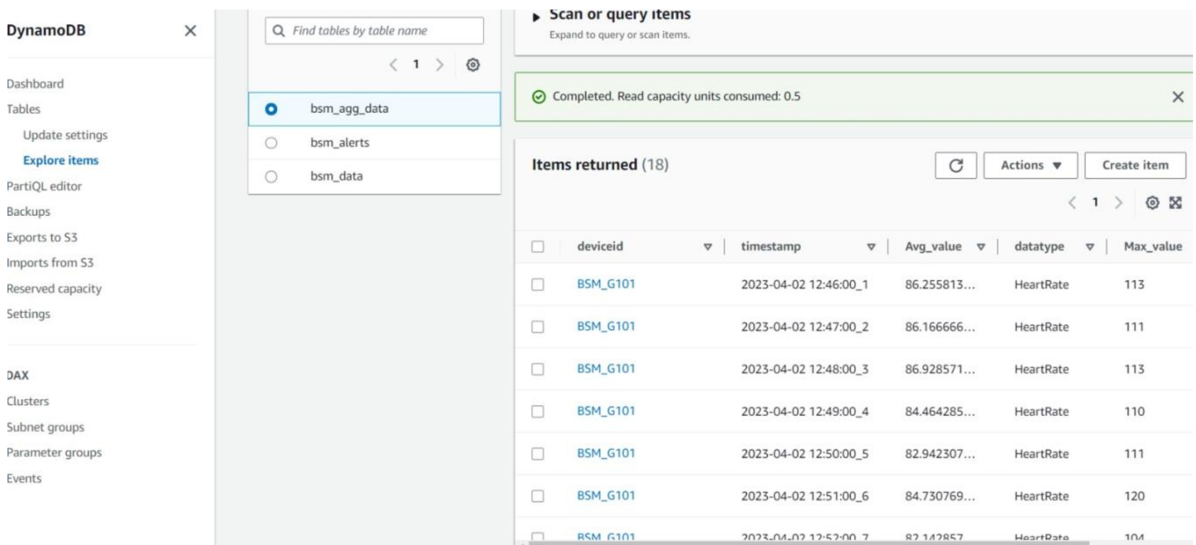
RAW DATA MODULE

In raw table whatever the data run in noted times , the data will store in in this module in seconds manner Provide functionality to aggregate the data from the three sensors individually, at the minute level. Should calculate and store the average, minimum and maximum values taking all values within a minute in account. Since we are only mimicking a continuous monitoring system, please provide Provide functionality to aggregate the data from the three sensors individually, at the minute level. Should calculate and store the average, minimum and maximum values taking all values within a minute in account. Since we are only mimicking a continuous monitoring system, please provide the aggregate or method call to be for a specific time range and aggregate



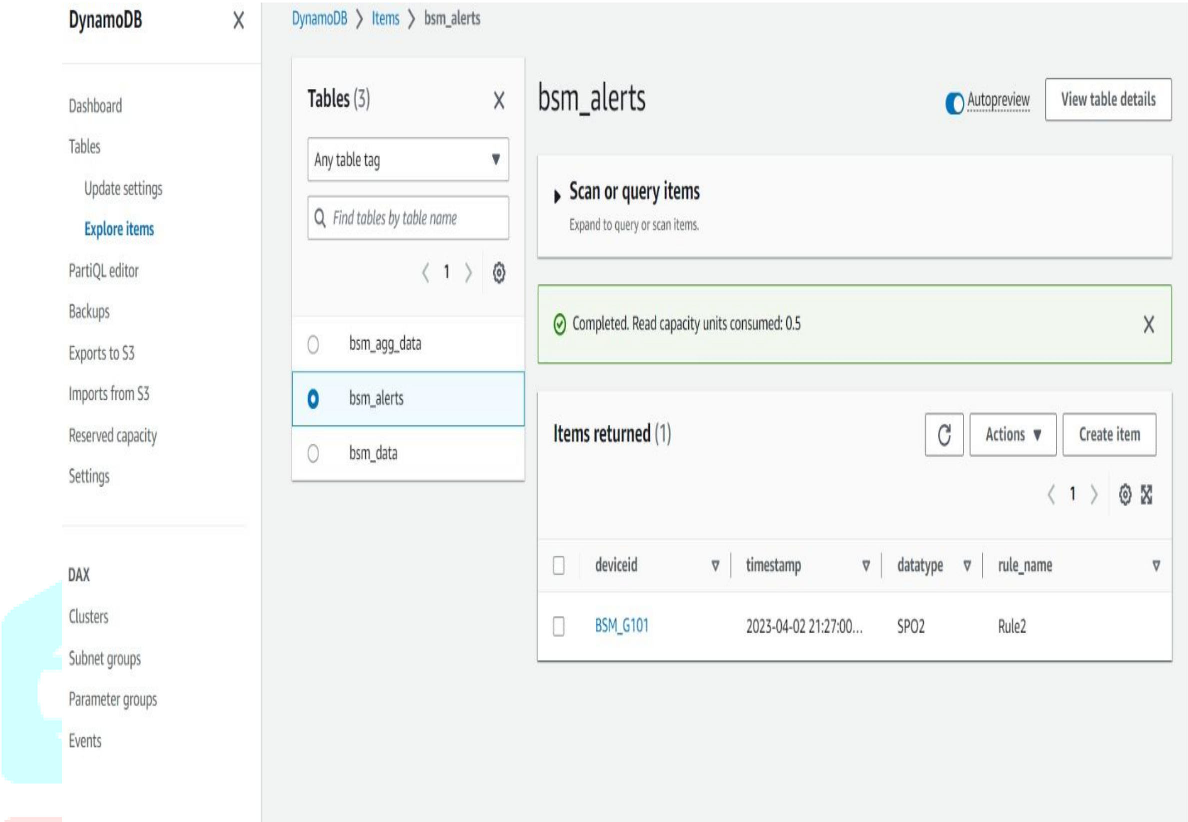
AGGREGATEMODULE

In aggregate table the data run in raw table will convert by one minutes and aggregate the results and the result data were store in this table



ALERT DATA MODULE

In alert table the exceed data or wrong data we restored in this table, and the data were store in this table will get a notification message by the user.



RawData Model.py

This file will process the raw data that was created and pushed in the bsm_raw_datatablebyBedSideMonitor.py.

AggregateData Model.py

This file will have implementation related to the aggregation on the raw data.

Database.py

This is a class that will invoke and call the various databases and methods to fetch and store the data in the bsm_agg_datatable.

AlertData Model.py

This is a file that will access the created rules and will detect the anomaly values based on the parsed rule. The detected output will then be pushed in the bsm_alerts.

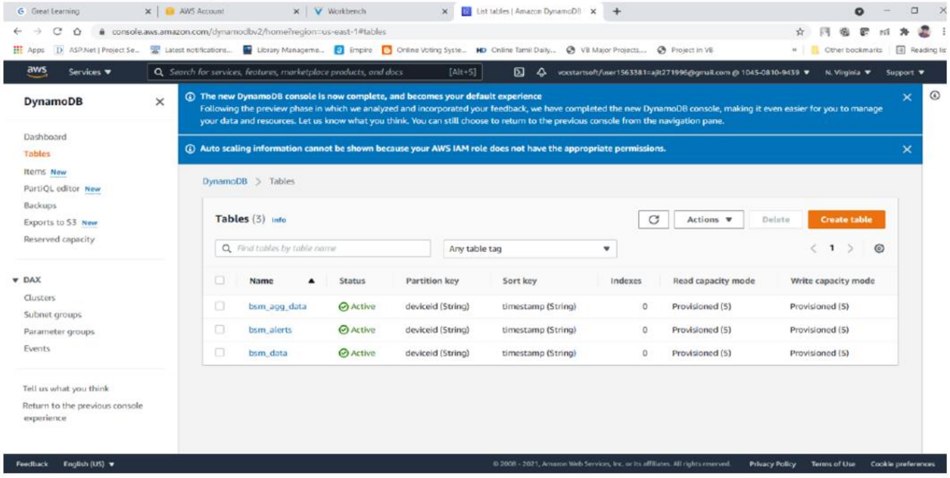
Main.py

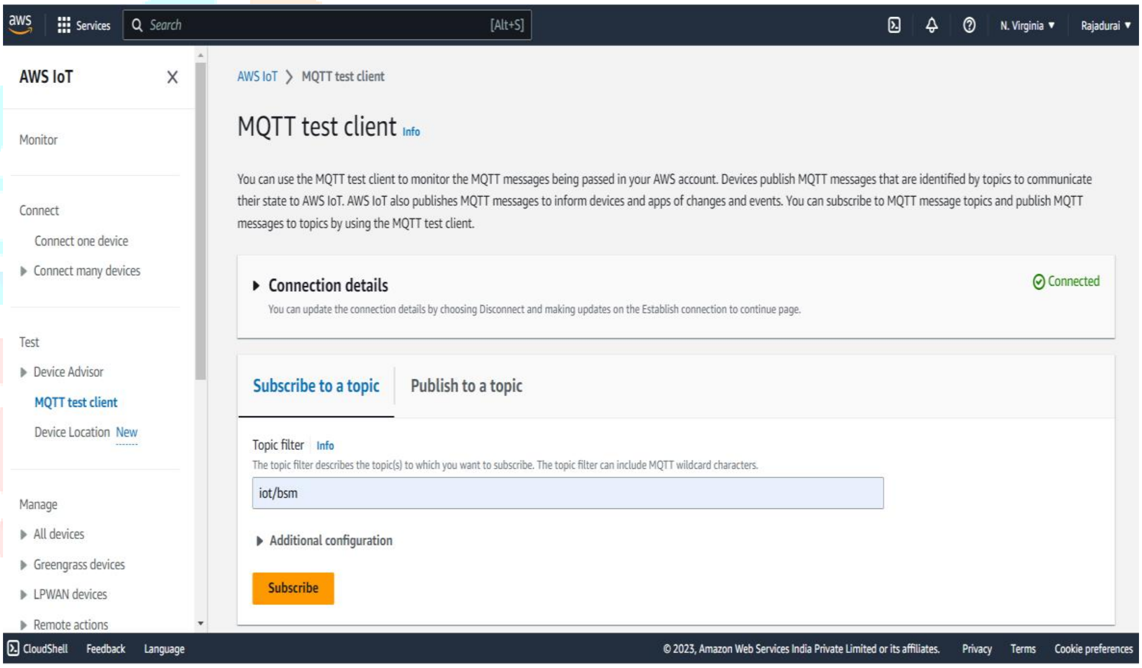
This will be the driver code that should be invoked to perform the aggregation operation and alert operation on the inserted data that is available in the raw form in one of the tables.

BedSideMonitor.py

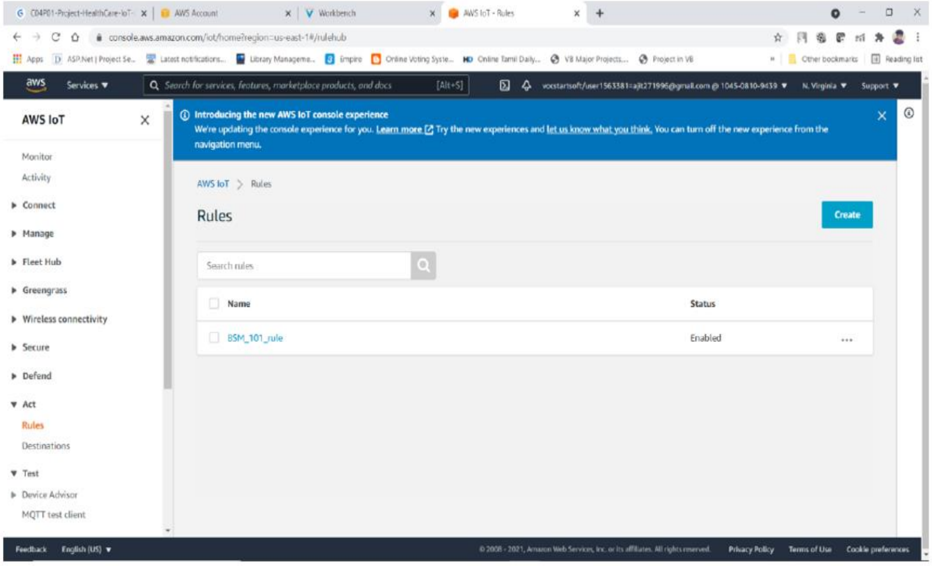
The modified or newly created data publisher should also be present in the zipped folder along with other files.

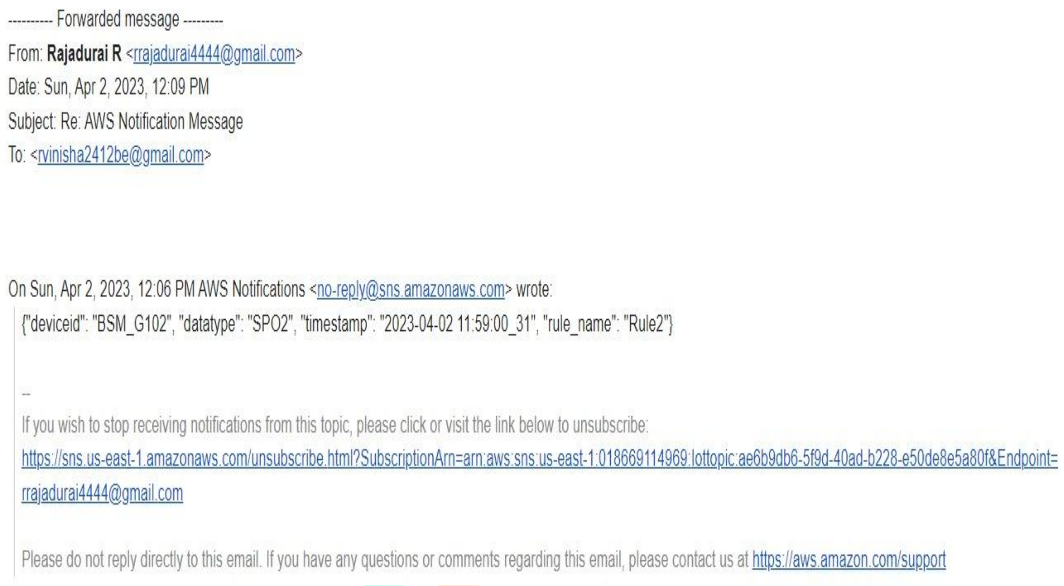
DYNAMO DATABASE TABLE





RULE CREATION TO PUSH RAW DATA





---------- Forwarded message ---------
From: **Rajadurai R** <rrajadurai4444@gmail.com>
Date: Sun, Apr 2, 2023, 12:09 PM
Subject: Re: AWS Notification Message
To: <rvinisha2412be@gmail.com>

On Sun, Apr 2, 2023, 12:06 PM AWS Notifications <no-reply@sns.amazonaws.com> wrote:

{"deviceid": "BSM_G102", "datatype": "SPO2", "timestamp": "2023-04-02 11:59:00_31", "rule_name": "Rule2"}

--
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-1:018669114969:lottopic:ae6b9db6-5f9d-40ad-b228-e50de8e5a80f&Endpoint=rrajadurai4444@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

V.CONCLUSION

In amazon web service we have to store the values of heart rate,spo2,body temperature and the values are stored in three different tables raw table ,aggregate table, anomaly table. In raw table we get data for everyone second and in aggregate table we get data for every one minute and it will given filtering the data by average, maximum, minimum . In exceed data or any other wrong data will best ordinal normally table. This process will helpful for the doctors to maintain patient health ,and we get notification message in E-mail if wrong data or exceed values occurs.

VI.FUTUREENHANCEMENT

In future we have plan that patient id that directly connected with a sensor and we get data for every one minute. Create a Better Time Efficient data will be stored in AWS dynamo db Increased Scalability for variable patient id and Enable Bulk of data stored. The ongoing trends in this domain have inspired healthcare institutions to implement IoT in their business operation models. Almost 60% of the businesses in the healthcare industry have already adopted IoT-based solutions in their infrastructure. It shows that companies and healthcare providers are looking for better solutions that rectify the issues they faced with manual record processing and management systems. Another report states that 80% of clinics prefer Electronic Health Records over manual documents. These advanced systems help hospitals store, manage, and analyze patient results more efficiently, leading to the replacement of manual record-keeping practices.

VII.ACKNOWLEDGMENT

We would like to acknowledge our sincere thanks to the Management of our College and our family members who have supported and helped us in different stages of this project work

VIII.REFERENCES

[1]-Gera et al. A patient health monitoring system that is built on IoT technology and is connected to the Cloud Talk platform

[2]- Wu, Wu Developed a small wearable sensorpatch that can assess a variety of physiological signals. Uses a smart phone as the mobile gateway, Raspberry Pi

[3]- Islam, Rahaman Proposed a real-time IoT system to monitor patients' vital signs and the room's environmental conditions..

[4]-Al-Sheikh and Ameen Designed an IoT healthcare-monitoring system that uses a mobile phone by sending data to specific smart phone. HR, SpO2,ECG,andBTIEEE02.11-System uses two microcontrollers.- Wi-Firange

[5]- Hamim, Paul Developed a prototype of IoT-based remote health-monitoring system