Smart Home Health System Using AWS Cloud Services

Ali Koteich
Email: ali.w.koteich@gmail.com

Fatimah AlZahraa Ezzeddine
Email: fatimahezz16@gmail.com

***Abstract*—Home health monitoring with IoT is reshaping healthcare, offering personalized and accessible solutions within homes. Through smart devices and wearables, individuals can seamlessly collect and transmit real-time health data, empowering proactive self-care. In this paper, we utilize the integration of IoT and cloud technologies to create an efficient and secure home health monitoring system. Leveraging an ESP-8266 board and a DHT-11 sensor [1] for temperature and humidity data collection (Figure 1), the system utilizes AWS services for seamless data transmission to the cloud. The collected data is then stored in an online database, providing a centralized repository accessible for analysis and visualization. This comprehensive approach enhances the capabilities of home health monitoring, enabling real-time insights into the indoor environment. The paper discusses the system architecture, implementation details, and potential applications, emphasizing the significance of cloud-based IoT solutions in advancing personalized and effective home healthcare.**

I. Introduction

With ongoing technological advancements, the potential for innovation is becoming evident, and one such imminent development is the Internet of Things (IoT). This concept is rapidly emerging as a ubiquitous global computing network, connecting everyone and everything to the Internet. The evolution of IoT is an active area of research with limitless possibilities, poised to reshape the current internet landscape into a more integrated and modified version. The increase of devices accessing internet services is on a constant rise, and connecting them all, whether through wired or wireless means, will create a robust source of information readily accessible. While the idea of enabling interaction between intelligent machines is cutting-edge, the technologies constituting IoT are not entirely novel to us. As implied by its name, IoT involves consolidating data from various things and channeling it to any virtual platform within the existing Internet infrastructure [2].

In the ever-evolving landscape of Internet of Things (IoT) applications, the project at hand introduces an innovative solution utilizing the ESP-8266 board[3]. This project focuses on real-time environmental monitoring by harnessing the capabilities of the ESP-8266 and a DHT-11 sensor. Specifically, the system is designed to capture crucial data points such as humidity and temperature. However, this is more than just a sensor system – it's an end-to-end IoT solution.

The collected environmental data is transmitted to the Amazon Web Services (AWS) Cloud [4], a robust and scalable cloud computing platform. Once in the cloud, the data finds its home in DynamoDB, an efficient NoSQL database service provided by AWS. This integration ensures that the data is not only securely stored but also accessible for further analysis, visualization, and integration into broader IoT applications (Figure 2).

This project encapsulates the spirit of IoT innovation, demonstrating the convergence of hardware (ESP-8266 and DHT-11) and cloud technologies (AWS) to create a cohesive system for environmental data monitoring and management. As we delve into the details of this project, we uncover a synergy that exemplifies the potential of IoT in seamlessly connecting the physical and digital realms for enhanced data insights and intelligent decision-making.

II. Methodology

A. IoT Hardware

The ESP8266 is a low-cost, Wi-Fi-enabled microcontroller developed by Espressif Systems. It gained popularity for its compact size, low power consumption, and built-in Wi-Fi capabilities. The ESP8266 allows developers to easily add wireless connectivity to various electronic projects, making it a popular choice for IoT (Internet of Things) applications. It features a built-in TCP/IP stack, making it capable of connecting to the internet and exchanging data. Additionally, it can be programmed using the Arduino IDE, making it accessible to a wide range of developers for creating connected devices and applications.

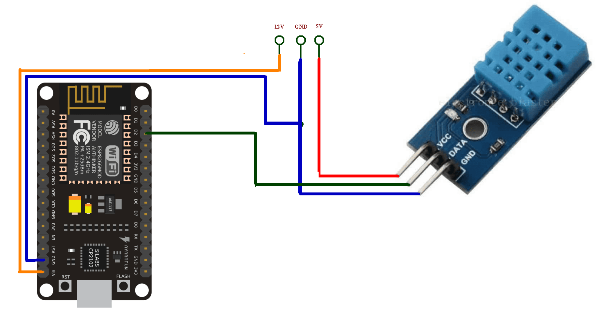

Fig. 1. ESP8266 and DHT11 connectivity.

1) ESP8266 and Sensors: By interfacing the ESP-8266 with a DHT-11 sensor that integrates the temperature and humidity sensing capabilities. DHT-11 sensor has three pins:

VCC, GND and Data, VCC is connected to a 3.3V output pin on the ESP8266, GND to ground, and Data to a digital GPIO pin. Then temperature and humidity data will be recorded every 2000 ms.

2) MQTT Protocol: Utilizing the ESP8266 in conjunction with the MQTT (Message Queuing Telemetry Transport) protocol provides compelling advantages for IoT applications [5]. The ESP8266, known for its resource efficiency, integrates effectively with the lightweight design of MQTT, ensuring minimal impact on the device's resources. With MQTT's publish-subscribe model, the ESP8266 can efficiently publish and subscribe to topics, enabling flexible and scalable communication. Asynchronous communication and varied Quality of Service (QoS) levels enhance real-time updates and message reliability on the ESP8266. The protocol's features, such as retained messages for storing the last known good value, cater to the ESP8266's functionality. MQTT's reliability over unreliable networks and scalability align well with the ESP8266's capabilities. Its wide adoption and support across platforms bring interoperability to the ESP8266 in IoT ecosystems. The security features of MQTT, including authentication and encryption, enhance communication integrity for the ESP8266. The broker-based architecture further simplifies message management, streamlining direct communication between ESP8266 devices. In essence, opting for MQTT with the ESP8266 establishes a robust, scalable, and efficient communication framework tailored to the specific requirements of IoT applications.

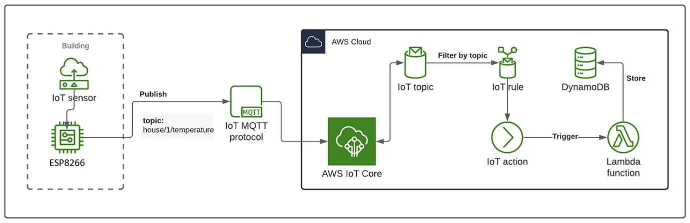

Fig. 2. Model's Architecture.

B. AWS Cloud Computing

Amazon Web Services (AWS) has emerged as a dominant force in the realm of cloud computing due to its compelling advantages. With a scalable and reliable infrastructure, AWS allows users to dynamically adjust resources to optimize performance. Its global network of data centers enables the deployment of applications close to end-users, minimizing latency. The extensive service portfolio encompasses computing power, storage, databases, machine learning, and analytics, catering to diverse business needs. AWS prioritizes reliability through multiple availability zones and data centers, ensuring high availability and minimal downtime. Security is a top priority, with AWS offering comprehensive tools and features to secure data, identity, and applications while adhering to compliance standards. The pay-as-you-go model makes AWS cost-effective and accessible to businesses of all sizes. Furthermore, AWS fosters constant innovation, provides developer-friendly tools, and has a supportive community and partner ecosystem, solidifying its position as a preferred cloud computing platform for technological advancement and business growth.

1) AWS IoT Core [6]: AWS IoT Core is a managed service by Amazon Web Services (AWS) that enables secure and scalable management of IoT devices. It allows for the secure connection, communication, and processing of data from IoT devices in the cloud. With features like device authentication, data processing rules, and device management capabilities, AWS IoT Core simplifies IoT device management and enables integration with other AWS services for data analytics and automation.

To set up AWS IoT Core with an ESP8266, we start by creating an AWS account and navigating to the AWS IoT Console. Create a new IoT Thing, generate security certificates, and download the certificate, private key, and Amazon Root CA file. Attach a policy to grant necessary permissions and note the Thing Name. In the Arduino IDE, we install the ”PubSubClient” library and write a sketch specifying our Wi-Fi credentials, AWS IoT Core endpoint, and certificate/key details. We upload the sketch to the ESP8266, connect it to our computer, and monitor the serial output to ensure successful connections to Wi-Fi and AWS IoT Core.

2) AWS Lambda [7]: To establish an effective integration between AWS IoT Core and DynamoDB we use AWS Lambda. We begin by setting up AWS IoT Core, create IoT Things for our ESP8266, and configure the policies and certificates to ensure secure communication. Next, we configure an IoT Rule in the AWS IoT Console to capture messages from our ESP8266 board and trigger a Lambda function. In the AWS Lambda Console, we create a new Lambda function, specifying the runtime and implementing the necessary logic to process incoming temperature and humidity data. Within the Lambda function, we leverage the AWS SDK to interact with DynamoDB and ensure that our Lambda function has the appropriate permissions by attaching a role that grants write access to DynamoDB, then we create a DynamoDB table to store the processed data, configure the IoT Rule action to invoke the Lambda function and test the setup by publishing a message from an IoT device to AWS IoT Core, observing the Lambda function's execution and data storage in DynamoDB. This architecture enables a robust flow of data from IoT devices to DynamoDB through the intermediary processing power of AWS Lambda, facilitating real-time data processing and persistence.

3) AWS DynamoDB [8]: Storing data in DynamoDB, AWS's fully managed NoSQL database service, involves several key considerations. As IoT messages are processed by the Lambda function, the relevant data is extracted and prepared for storage. DynamoDB, being a highly scalable and low-latency database, offers a schema-less structure that accommodates the dynamic nature of IoT data. The Lambda function constructs parameters to insert and update items in a DynamoDB table, typically created beforehand to suit the data schema. The chosen table is designed with key attributes which is the timestamp, temperature and humidity data. This data can be carried for further data analytics tasks by using data analytics tools and visualization graphs.

C. AWS and Security

Employing Amazon Web Services (AWS) for cloud computing establishes a robust platform characterized by scalability and security. Services like AWS IoT Core facilitate the smooth integration and communication between IoT devices and the cloud. Security protocols within AWS IoT encompass device authentication and encryption, guaranteeing the confidentiality and integrity of data throughout its transmission.

AWS Identity and Access Management (IAM) affords precise control over access permissions, augmenting security within the IoT ecosystem. Furthermore, AWS Key Management Service (KMS) provides a secure avenue for the management of encryption keys, ensuring the protection of data stored in the cloud.

The AWS cloud infrastructure centralizes the management and monitoring of IoT devices, fortifying real-time threat detection capabilities. Noteworthy security features, such as AWS IoT Device Defender, ensure continuous monitoring and alerting, bolstering the resilience of IoT deployments against potential threats.

By harnessing AWS for cloud computing in IoT undertakings, organizations gain access to a comprehensive suite of security tools and services. AWS's unwavering commitment to compliance certifications and ongoing advancements in security technologies underscores its dedication to furnishing a secure and scalable environment for IoT implementations. The integration of AWS services guarantees the optimization of IoT project potential without compromising data security and the overall integrity of the system.

III. Conclusion

In summary, this article has explored the utilization of the ESP8266 microcontroller and DHT11 sensor to gather environmental data. By integrating these devices with AWS IoT Core, a robust IoT platform, we have established an efficient data transmission process. AWS Lambda functions were utilized to process and transform the data before storing it in DynamoDB. This end-to-end solution not only demonstrates the potential of IoT technology but also highlights the effectiveness of cloud-based services in facilitating streamlined data workflows. This comprehensive approach showcases the transformative possibilities when combining embedded systems, cloud computing, and serverless architectures. As hardware and cloud technologies converge, the framework presented in this article exemplifies innovation and scalability in modern IoT applications.

Future Work

In the realm of future work, there are several promising avenues for exploration. Firstly, expanding the range of sensors to include additional environmental parameters, such as air quality or light intensity, would enhance the system's comprehensiveness. Secondly, optimizing communication protocols and strengthening error-handling mechanisms would contribute to a more robust and reliable data flow. Thirdly, exploring the capabilities of edge computing for data processing closer to the data source holds potential for reducing latency. Fourthly, addressing security and privacy concerns through encryption and access controls is crucial for strengthening the system. Lastly, evaluating and improving scalability to accommodate a growing number of devices and data points remains a key priority for future advancements. These directions collectively aim to enhance the system's functionality, efficiency, and adaptability in the evolving landscape of IoT technologies.

References

[1] https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.

[2] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A review on internet of things (iot)," *International journal of computer applications*, vol. 113, no. 1, pp. 1–7, 2015.

[3] J. Mesquita, D. Guimarães, C. Pereira, F. Santos, and L. Almeida, "Assessing the esp8266 wifi module for the internet of things," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 784–791.

[4] https://aws.amazon.com/.

[5] D. Soni and A. Makwana, "A survey on mqtt: a protocol of internet of things (iot)," in *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, vol. 20, 2017, pp. 173–177.

[6] https://aws.amazon.com/iot-core/.

[7] https://aws.amazon.com/lambda/.

[8] https://aws.amazon.com/dynamodb/.