Real-Time IoT Data Visualization and Monitoring Dashboard Using AWS Kinesis and QuickSight

Developer: Oyekanmi Lekan

Github Code: https://github.com/intellisenseCodez/IoT-Monitoring-Dashboard

Problem Statement

xyz limited needs an efficient, real-time, serverless, and cost-effective AWS-native solution to monitor energy consumption in an apartment, connected to an IoT box.

As a Junior **Data Engineer**, you are required to:

1. **Design an architecture diagram** for the solution.
2. **Write Terraform code** to deploy the solution.
3. **Document a framework** detailing the solution's design, implementation, and operational considerations.
4. **(Bonus)** Include a data batching solution to optimize performance and cost.

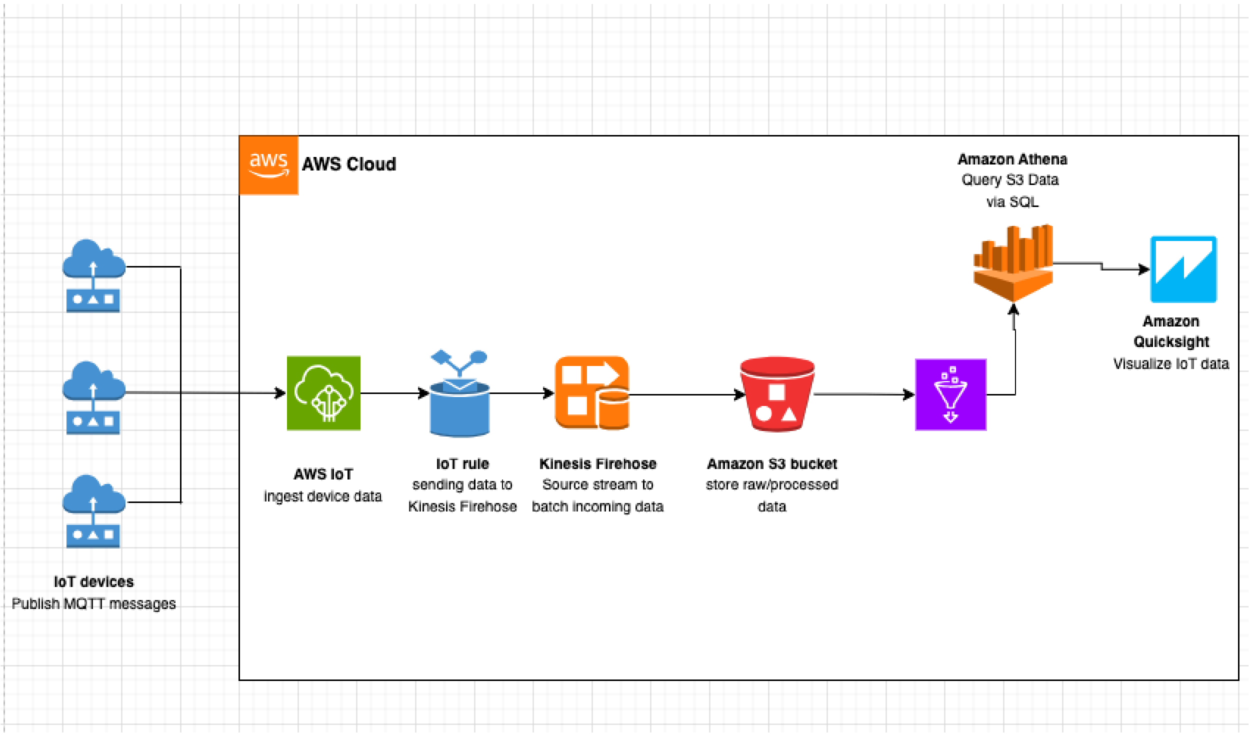
1: Understand Requirements

Before jumping to design, it’s best to understand the underlying requirements.

Requirement	Goal
Cost-effectiveness	Use serverless & managed services (Kinesis Firehose, S3, Athena)
Security	Encrypt data in transit (TLS) and at rest (SSE, IAM, secure MQTT)
Low Latency	Near real-time streaming via Kinesis and AWS IoT Core
High Availability	Use highly available AWS managed services

2. Proposed Solution

The solution is designed to provide a real-time, serverless, and cost-effective monitoring system for energy consumption in an apartment, connected to an IoT box. It uses a combination of AWS services such as AWS IoT Core, Amazon Kinesis, AWS Lambda, Amazon S3, AWS Glue, Amazon Athena, and Amazon QuickSight to achieve the desired functionality. The architecture is serverless and designed for scalability, cost optimization, and high availability.



Github Code: https://github.com/intellisenseCodez/IoT-Monitoring-Dashboard

4. AWS service overview

1. **IoT Core**: The energy consumption data is generated by IoT devices and is sent to AWS IoT Core. IoT Core facilitates the ingestion of data securely from IoT devices and acts as a messaging hub for the incoming data.

2. **Kinesis Data Streams**: The data from IoT Core is then routed through Amazon Kinesis Data Streams to provide real-time processing and analytics capabilities. Kinesis is a fully managed, scalable, and reliable service that can ingest and process large amounts of real-time data.

3. **AWS Lambda (Optional)**: Lambda functions can be used to process or transform the incoming IoT data before sending it to Kinesis or storing it in Amazon S3. Lambda ensures that processing is event-driven and serverless, removing the need to manage

infrastructure.

4. **Amazon S3**: The raw data from Kinesis (or processed data from Lambda) is stored in Amazon S3 for further analysis. S3 serves as a durable, scalable, and low-cost storage solution.

5. **AWS Glue**: AWS Glue is used to catalog the data in Amazon S3 and provide ETL (extract, transform, load) capabilities. The Glue Data Catalog provides a central repository for all the data and makes it available for querying in Amazon Athena.

6. **Amazon Athena**: Athena allows you to run SQL queries directly on the data stored in Amazon S3 without needing to provision or manage infrastructure. This service is used to analyze the energy consumption data efficiently.

7. **Amazon QuickSight**: QuickSight is used for visualizing the data stored in Amazon S3 and queried through Athena. QuickSight provides interactive dashboards and reports for monitoring energy consumption and trends over time.

8. **Amazon Kinesis Firehose (Optional)**: If necessary, Amazon Kinesis Firehose can be used to stream the processed data directly into S3 or to other downstream services.

Security Measures Implemented

1. **Data Encryption**:
   In Transit: Data transmitted from IoT devices to AWS IoT Core, as well as between services (Kinesis, Lambda, S3, etc.), is encrypted using TLS (Transport Layer Security) to ensure the confidentiality of sensitive data.

2. **IAM Roles and Policies**:

   AWS Identity and Access Management (IAM) roles and policies are configured to ensure least-privilege access. Each service only has the permissions necessary to perform its specific tasks (e.g., IoT Core only sends data to Kinesis).

Cost Optimization, Scalability, and Fault Tolerance

1. **Serverless Architecture**: The solution leverages serverless AWS services such as IoT Core, Lambda, Athena, and QuickSight, which only charge for actual usage. This eliminates the need to manage and provision servers, reducing costs.
2. **S3 Storage**: Storing data in Amazon S3 is cost-effective, especially when using data formats like Parquet or ORC that reduce storage costs.

3. **Kinesis**: Kinesis automatically scales to accommodate varying data volumes. It adjusts throughput according to the incoming data stream, ensuring that the system can handle large volumes of energy consumption data without manual intervention.
4. **Lambda**: Lambda automatically scales based on the incoming event load. It handles bursts of data seamlessly and scales down when not in use, making it highly cost-efficient.
5. **QuickSight**: QuickSight scales automatically to accommodate large datasets and multiple users, allowing for real-time visualizations without compromising performance.
6. **Kinesis**: Kinesis offers built-in durability and fault tolerance. It stores data across multiple availability zones and automatically replicates data to ensure that it remains available even in the event of infrastructure failures.
7. **S3**: S3 is designed for high durability (99.999999999%) and provides fault tolerance by replicating data across multiple availability zones.
8. **QuickSight**: QuickSight is a fully managed service that ensures high availability and fault tolerance by operating across multiple regions and availability zones.