IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15327

BUILD A REAL – TIME ANALYTICS DASHBOARD WITH AWS KINESIS AND LAMBDA

DharshiniPriya.R[1], Dr.S. Thavamani[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore 641006, Tamil Nadu, India'

Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore-641006, Tamil Nadu, India[2]

Abstract: Real-time weather monitoring plays an important role in many sectors including agriculture, transportation, disaster management, and environmental monitoring. Traditional weather systems generally rely on periodic updates which may not provide immediate information when environmental conditions change rapidly. With the advancement of cloud computing and streaming technologies, modern applications are capable of processing and visualizing data continuously as events occur. This paper presents WeathX, a Real-Time Weather Analytics Dashboard that demonstrates a cloud-inspired streaming architecture for continuous weather monitoring and analytics. Weather data is collected automatically from the OpenWeatherMap API at regular intervals and processed through a simulated event-driven pipeline inspired by technologies such as Amazon Kinesis, AWS Lambda, and Amazon DynamoDB. Each weather record is treated as a streaming event and processed through ingestion, transformation, storage, and visualization layers. The backend system developed using Node.js manages data fetching, event processing, and communication with the frontend. The frontend dashboard built using React.js provides interactive visualization of weather parameters including temperature, humidity, pressure, and wind speed. WebSocket communication allows real-time updates without requiring manual page refresh. The proposed system demonstrates how real-time streaming architectures can support efficient analytics platforms and improve the responsiveness of weather monitoring applications.

Keywords: Real-Time Analytics, Weather Monitoring, Streaming Architecture, WebSockets, Cloud Computing

1.INTRODUCTION

Weather information is an essential component of many real-world applications. Accurate weather monitoring helps people plan daily activities and supports decision-making in agriculture, aviation, transportation, and disaster management. Conventional weather applications usually display data that is updated periodically, which means users may not always receive the most recent weather conditions. This limitation reduces the effectiveness of traditional monitoring systems when continuous real-time information is required.

The rapid growth of cloud computing and distributed systems has introduced new technologies that allow applications to process streaming data in real time. Event-driven architectures enable systems to react instantly to incoming data events and process information continuously rather than in scheduled batches. Streaming platforms such as Amazon Kinesis and Apache Kafka have made it possible to build scalable pipelines capable of handling large volumes of data.

The WeathX Real-Time Weather Analytics Dashboard is designed to demonstrate how such streaming architectures can be applied to weather monitoring systems. The system integrates data collection, processing, storage, and visualization into a unified platform. By using modern web technologies and a simulated cloud-based pipeline, the system provides continuous weather updates through an interactive dashboard interface.

2. LITERATURE SURVEY

Several weather monitoring platforms have been developed to provide weather information to users through web and mobile applications. Many traditional systems rely on centralized data sources and periodic polling mechanisms to retrieve weather updates. Although these systems provide useful information, they often lack real-time processing capabilities and interactive visualization features.

DOI: 10.17148/IJARCCE.2026.15327

Recent research has focused on the use of streaming data architectures for real-time analytics applications. Technologies such as Apache Kafka, Amazon Kinesis, and Apache Storm allow systems to process large volumes of data streams efficiently. These technologies are widely used in financial systems, social media platforms, and environmental monitoring systems where continuous data processing is required.

In addition to streaming frameworks, modern visualization technologies such as React.js and data visualization libraries enable developers to build interactive dashboards that display data dynamically. Real-time communication protocols like WebSockets allow servers to push updates directly to clients without requiring repeated requests. The proposed WeathX system combines these concepts to build a real-time analytics platform for weather monitoring.

3. METHODOLOGY

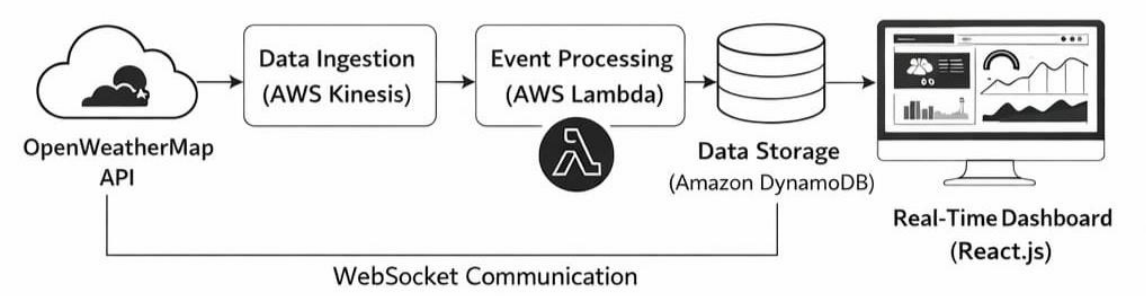
The WeathX system follows a modular architecture that simulates a real-time streaming pipeline. The methodology includes multiple layers responsible for data collection, data ingestion, processing, storage, and visualization. Each component of the system performs a specific task while interacting with other modules to maintain a continuous data flow.

3.1 Architectural Design

The architecture of the system is inspired by modern cloud-based streaming platforms. The first component of the architecture is the data collection layer, which retrieves weather information from the OpenWeatherMap API. The collected data includes parameters such as temperature, humidity, pressure, wind speed, and weather conditions.

After data collection, the weather records are pushed into a simulated ingestion pipeline that represents a streaming platform similar to Amazon Kinesis. This layer treats each weather record as a streaming event and forwards it to the processing module.

The processing layer is inspired by AWS Lambda and automatically processes incoming events. The module extracts relevant parameters from the raw JSON data, formats the data structure, and attaches metadata such as timestamps. The processed data is then stored in an in-memory datastore that simulates Amazon DynamoDB.



System Architecture of the WeathX Real-Time Analytics Dashboard

3.2 Workflow / System Flow

The workflow of the system begins with the automatic retrieval of weather data from the OpenWeatherMap API at predefined intervals. The retrieved data is then pushed into the ingestion pipeline where it is treated as an event in the streaming system.

The processing module receives these events and transforms the raw weather data into a structured format suitable for storage and visualization. Once processed, the data is stored in the datastore and made available through backend APIs.

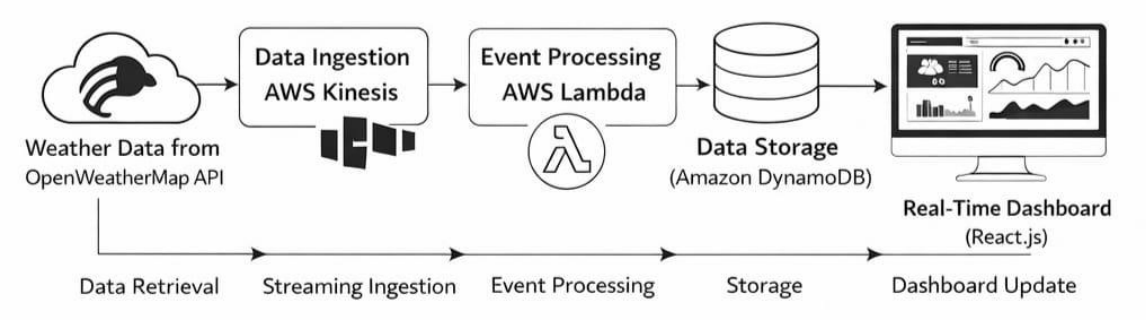
The frontend dashboard communicates with the backend using REST APIs and WebSocket connections. Whenever new weather data is processed, the server pushes updates to the dashboard interface, allowing users to view real-time weather analytics without refreshing the page.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15327



Workflow of the real-time weather data pipeline illustrating data retrieval,streaming ingestion,processing,storage and dashboard updates

Table 1: Technology Stack Used in the System

Component	Technology Used	Purpose
Frontend	React.js	Dashboard interface and visualization
Backend	Node.js	Server-side processing
API	OpenWeatherMap API	Fetch real-time weather data
Streaming Simulation	Node.js Modules	Simulate Kinesis streaming
Processing	Lambda-like Function	Event processing
Data Storage	In-memory datastore	Simulate DynamoDB
Communication	WebSocket	Real-time updates

4. RESULTS AND EXPLANATION

The implemented system successfully demonstrates the use of a streaming architecture for real-time weather monitoring. The dashboard displays multiple weather parameters including temperature, humidity, wind speed, and atmospheric pressure for selected cities.

Interactive charts allow users to observe weather trends over time, while pipeline statistics provide insights into system performance. The use of WebSocket communication ensures that data updates are delivered instantly to the frontend.

Testing results show that the system effectively handles continuous weather data retrieval and processing while maintaining stable performance. The modular design also allows easy extension of system functionality.

The dashboard also provides insights into system performance by displaying pipeline statistics such as the number of processed events and recent weather records. These metrics help evaluate the efficiency of the streaming architecture and ensure that data flows smoothly through each processing stage. The results demonstrate that the event-driven design allows the system to respond quickly to incoming weather data while maintaining a responsive user interface.

5. CONCLUSION AND FUTURE SCOPE

The WeathX Real-Time Weather Analytics Dashboard demonstrates how streaming architectures can be used to build efficient real-time monitoring systems. The system integrates data collection, event processing, storage, and visualization within a unified architecture.

The project highlights the advantages of event-driven processing and real-time communication technologies in modern web applications. By using a simulated cloud environment, the system demonstrates how scalable streaming platforms operate.

Future improvements may include integration with real cloud services such as Amazon Kinesis, AWS Lambda, and DynamoDB. Additional features such as machine learning based weather prediction, mobile application support, and real-time alert notifications could further enhance the capabilities of the system.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15327

REFERENCES

[1]. OpenWeatherMap API Documentation. https://openweathermap.org/api
[2]. React.js Official Documentation. https://react.dev
[3]. Node.js Documentation. https://nodejs.org
[4]. Amazon Web Services. *Amazon Kinesis Developer Guide*.
[5]. Amazon Web Services. *AWS Lambda Developer Guide*.
[6]. Amazon Web Services. *Amazon DynamoDB Developer Guide*.
[7]. Kreps, J., Narkhede, N., & Rao, J. (2011). *Kafka: A Distributed Messaging System for Log Processing*.
[8]. Zaharia, M., et al. (2013). *Discretized Streams: Fault-Tolerant Streaming Computation at Scale*.