

# Real-Time Weather Monitoring and Analytics using Big Data Technologies

**Name:** Abhilash Tarigopula

**Subject:** Big Data & Distributed Process (MCSCIN5A1825)

## Abstract

This project implements an end-to-end Big Data pipeline for real-time weather monitoring. Live weather data is collected from a Weather API, streamed using Kafka, processed with Apache Spark, stored using distributed storage systems, and visualized through Grafana dashboards.

## 1. Introduction

Weather data is generated continuously and changes rapidly. Traditional systems are not suitable for handling such streaming data at scale. This project demonstrates a scalable and distributed architecture for real-time weather analytics using Big Data technologies.

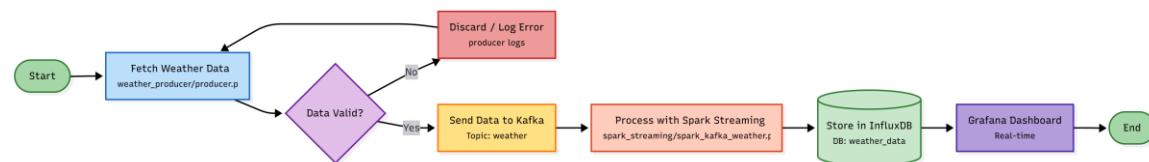
## 2. Weather Data Source (API)

The Weather API is the starting point of the pipeline. It provides live weather parameters such as temperature, humidity, and wind speed. Data is fetched periodically using a Python-based producer and converted into JSON format before being sent to Kafka.

## 3. System Architecture Overview

The architecture follows a standard Big Data pipeline:

Weather API → Kafka → Spark → InfluxDB → Grafana



In parallel, Spark stores historical data in HDFS for long-term analysis.

## 4. Technology Explanation

### Kafka

Kafka is used for real-time ingestion of weather data. It ensures fault tolerance and decouples data producers from consumers.

### Spark

Apache Spark processes streaming data received from Kafka. It performs transformation and supports distributed processing.

### HDFS

HDFS is used for long-term storage of historical weather data and enables batch analytics.

### InfluxDB

InfluxDB stores recent weather data optimized for time-series queries.

### Grafana

Grafana visualizes the weather data using live dashboards.

## 5. Cluster and Infrastructure Validation

Grafana Cluster Monitoring (CPU & Memory)



## Spark, Kafka, and Hadoop Services Status

## HDFS NameNode Web UI

localhost:9870/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

## Overview 'master:9000' (✓active)

Started:	Sun Nov 23 14:52:38 +0100 2025
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 10:22:00 +0200 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-453abb12-bb56-4a4a-8dcd-7df93083ac44
Block Pool ID:	BP-203575494-10.0.0.24-1761053081425

## Summary

Security is off.  
 Safemode is off.  
 74 files and directories, 40 blocks (40 replicated blocks, 0 erasure coded block groups) = 114 total filesystem object(s).

## YARN ResourceManager UI

localhost:18088/cluster

 A

**Cluster Metrics**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Memory
0	0	0	0	0	<memory:0 B, vCores:0

**Cluster Nodes Metrics**

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
2	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
Showing 0 to 0 of 0 entries										

## Spark Application UI

The screenshot shows the Apache Spark 3.5.7 Jobs page. At the top, there's a navigation bar with links to YouTube, Maps, Gmail, Credila, Usha, smartpay.buzzily.co..., python, Course: The Comple..., PyFormat: Using %..., CodingBat Python, and All. Below the navigation bar, there's a header with the Apache Spark logo and the text "Spark 3.5.7". The main content area is titled "Spark Jobs" with a question mark icon. It displays the following information:

- User: adm-msc
- Total Uptime: 14 min
- Scheduling Mode: FIFO

A link to "Event Timeline" is also present.

## Weather Grafana Dashboard

### Dashboard Output Explanation

The final output of this project is a **real-time Grafana dashboard** that visualizes weather data collected and processed through the Big Data pipeline. The dashboard provides an intuitive and user-friendly view of live weather conditions.

The data shown in the dashboard follows this flow:

**Weather API → Kafka → Spark → InfluxDB → Grafana**

Each panel in the dashboard represents a specific weather parameter and displays values over time.

---

### Live Wind Speed Panel

This panel displays the **wind speed** values received from the Weather API.

- The horizontal axis represents **time**
- The vertical axis represents **wind speed**
- Each point on the graph corresponds to a wind measurement stored in the database

This panel helps in understanding changes in wind conditions over time and confirms that streaming data is successfully ingested and visualized.

---

### Live Humidity Panel

This panel shows the **humidity percentage** in the atmosphere.

- Higher values indicate more moisture in the air
- Lower values indicate drier conditions

The humidity panel demonstrates that multiple weather parameters are processed correctly and stored as time-series data.

---

## Live Temperature Panel

This panel visualizes the **temperature in degrees Celsius**.

- Rising values indicate warming conditions
- Falling values indicate cooling conditions

This panel confirms that numeric weather data is continuously processed and displayed in real time.

---

## Interpretation of Data Points

Currently, the dashboard shows individual data points rather than continuous lines. This is expected behavior because:

- The system has received a limited number of data records
- Each point represents a successfully processed and stored weather reading

As more data is collected over time, these points naturally form continuous time-series graphs.

---

## Significance of the Dashboard

The dashboard proves that:

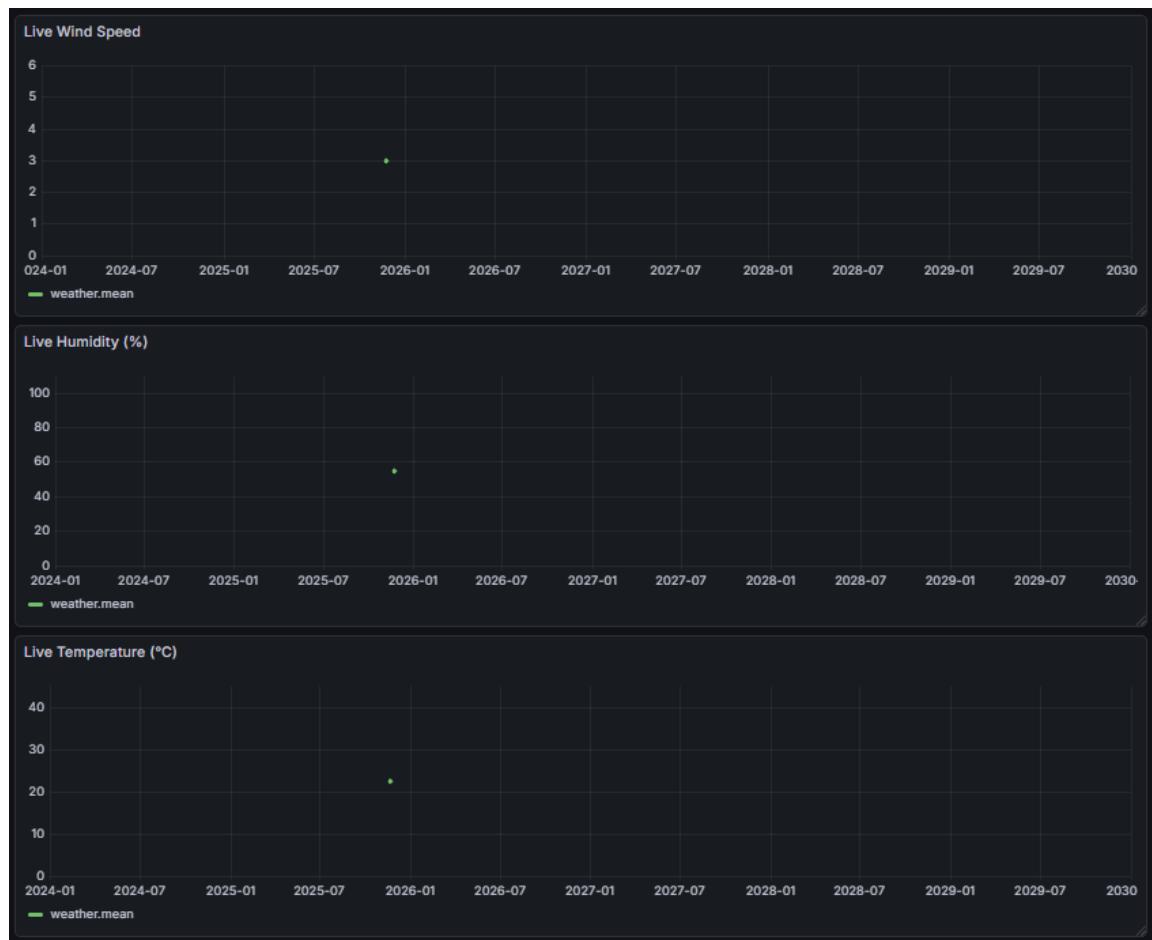
- Real-time data ingestion using Kafka is working
- Data processing and transformation are successful
- Time-series storage in InfluxDB is correctly configured
- Grafana is able to query and visualize live data

Overall, the dashboard serves as **visual proof of a fully functional end-to-end Big Data pipeline**.

---

## Summary

The Grafana dashboard provides real-time visibility into weather conditions and validates the successful integration of all Big Data components used in this project. It allows users to monitor live data easily and demonstrates the practical application of Big Data technologies in real-world scenarios.



## 6. Results and Output

The system successfully displays live weather data such as wind speed, humidity, and temperature. The Grafana dashboard updates automatically as new data arrives.

## **7. Conclusion**

This project demonstrates a complete Big Data pipeline for real-time weather monitoring. Kafka enables ingestion, Spark provides processing, HDFS supports historical storage, InfluxDB manages time-series data, and Grafana enables visualization.