

Project Report: Real-Time Weather Dashboard

GitHub Repository : [khandelwal-nancy/real-time-weather-project \(github.com\)](https://github.com/khandelwal-nancy/real-time-weather-project)

Project Overview

The Real-Time Weather Dashboard project is a data-driven solution designed to provide users with up-to-the-minute weather information and forecasts for multiple cities. Leveraging the power of Microsoft Azure services, this project seamlessly integrates weather data from multiple sources and delivers it through an interactive Power BI dashboard.

Objectives

The primary objective of this project is to offer a real-time weather tracking and forecasting tool that empowers users with critical weather insights for informed decision-making.

Key objectives include:

- Collecting real-time weather data from OpenWeatherMap API and 12-hour forecasts from Azure Maps API.
- Processing and transforming raw data into meaningful insights.
- Creating an interactive Power BI dashboard for end-users.
- Establishing an automated data pipeline for continuous updates.

Project Components

The project consists of the following components:

Data Collection

Current Weather Data

- The "EventHubProducer" notebook fetches current weather data from the OpenWeatherMap API.
- This data is processed and sent to Azure Event Hubs for further handling.

12-Hour Forecast Data

- The "ForecastDataLoad" notebook retrieves 12-hour weather forecast data from the Azure Maps API.

Data Processing

Current Weather Data

- Azure Stream Analytics captures and processes events from Azure Event Hubs.
- Filtered data is loaded into Azure SQL Database (SQL Server) for storage.

Forecast Data

- The "ForecastDataProcessing" notebook processes forecast data and stores it in Azure Data Lake Storage (ADLS).

Storage

Azure Data Lake Storage (ADLS)

- Processed forecast data is saved in ADLS as Parquet files, partitioned by timestamp

Azure SQL Database (SQL Server)

- Processed data is stored in the "dbo.weatherforecast" table in Azure SQL Database (SQL Server).

Automation

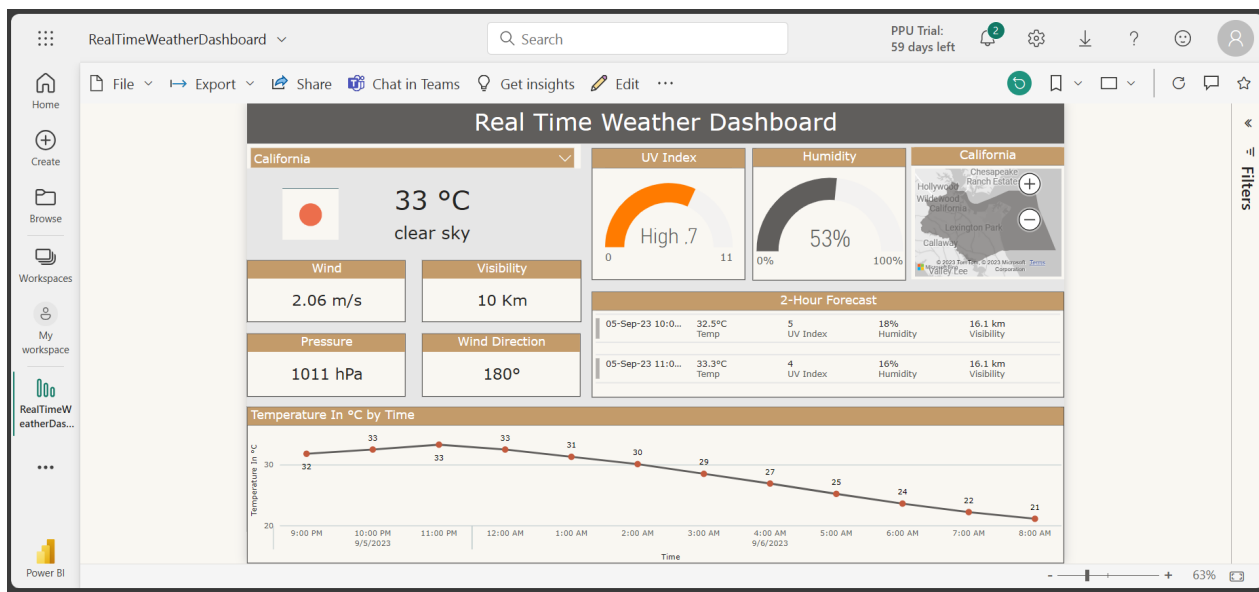
Azure Data Factory (ADF)

- ADF pipelines orchestrate notebook execution, ensuring regular data updates and synchronization.

Visualization

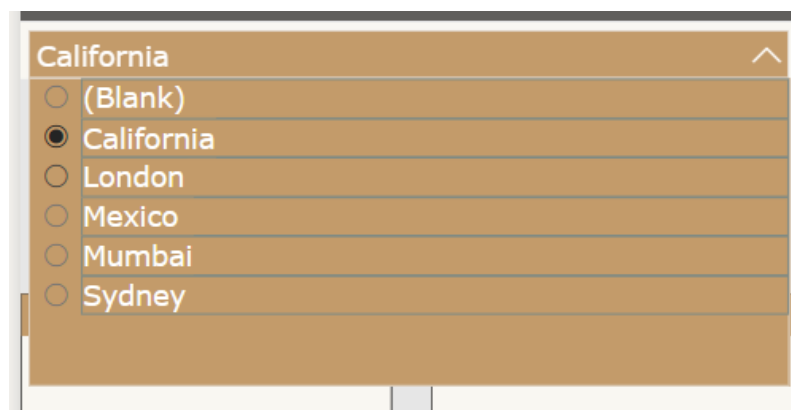
Power BI Dashboard

- The "Real-Time Weather Dashboard" in Power BI offers interactive visualizations of weather data.
- Visualizations include temperature, humidity, visibility, wind speed, UV index, wind direction, and pressure.

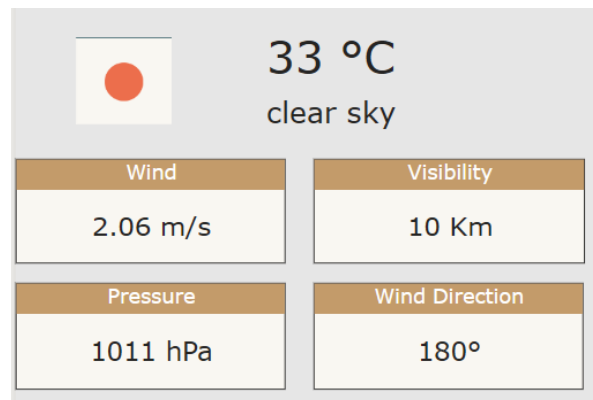


Components of Power Bi Dashboard

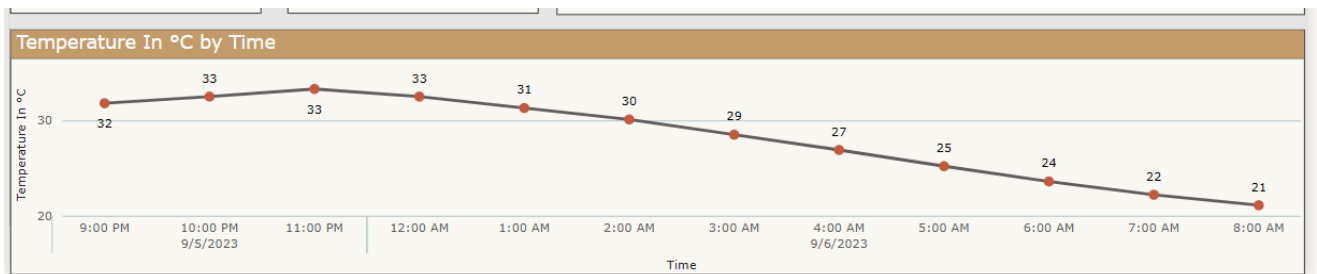
- Data Slicer - Users can filter data by selecting a specific city from the slicer, allowing for customized views.



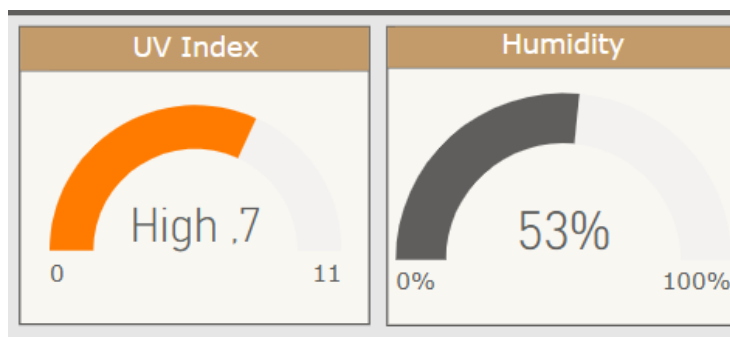
- Cards - Shows real-time weather information based on the city selected in Slicer.



- Line Chart - Line chart displaying 12-hour forecast for temperature.



- Gauges - UV Index Gauge provides real-time insights into sun exposure risk levels. The UV index is categorized into key zones: Low, Moderate, High and Extreme, each represented by a distinct color scheme for easy interpretation. Humidity gauge represents the latest percentage humidity.



- Multiple Cards - 2-Hour Detailed Weather Visualization in a single card.

2-Hour Forecast				
05-Sep-23 10:0...	32.5°C Temp	5 UV Index	18% Humidity	16.1 km Visibility
05-Sep-23 11:0...	33.3°C Temp	4 UV Index	16% Humidity	16.1 km Visibility

Impact and Use Cases:

The Real-Time Weather Dashboard project has diverse applications:

- Daily weather updates for personal use.
- Decision support for event planning, travel, or outdoor activities.
- Integration with other applications for contextual weather information

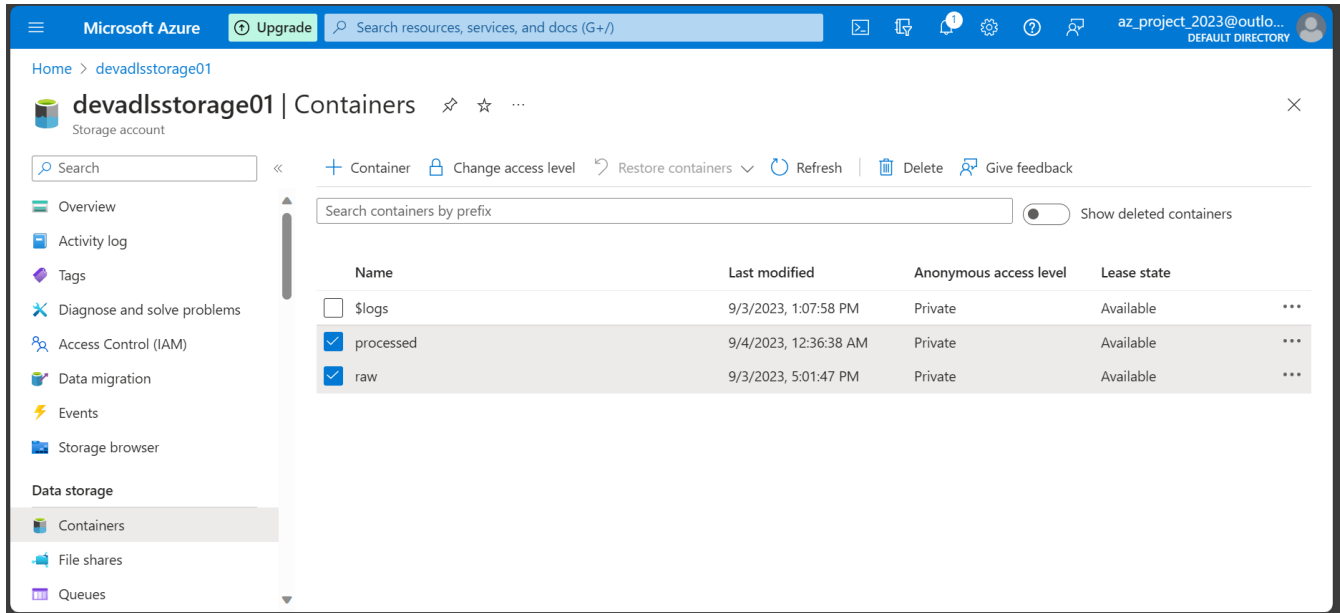
Conclusion:

The Real-Time Weather Dashboard project harnesses Azure's cloud capabilities to provide real-time weather insights to users. Its user-friendly interface and automated data updates make it a valuable tool for weather enthusiasts, travelers, and decision-makers. As it evolves, this project promises to become an indispensable resource for staying informed about weather conditions and forecasts.

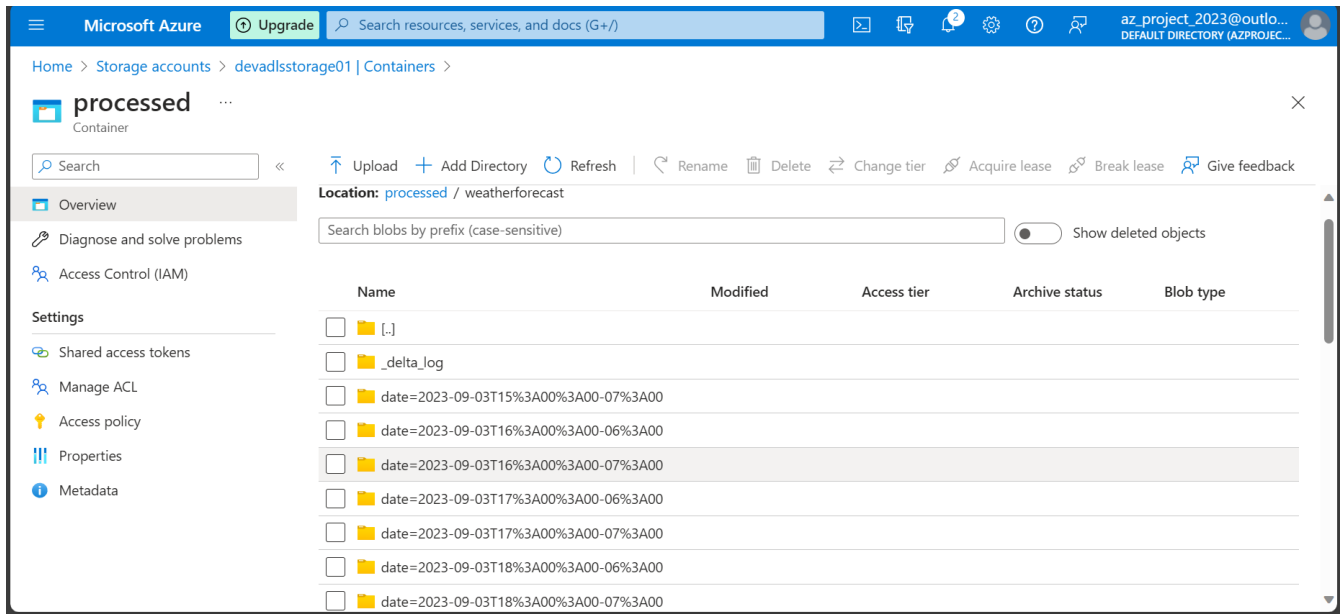
Appendix

A. Screenshots

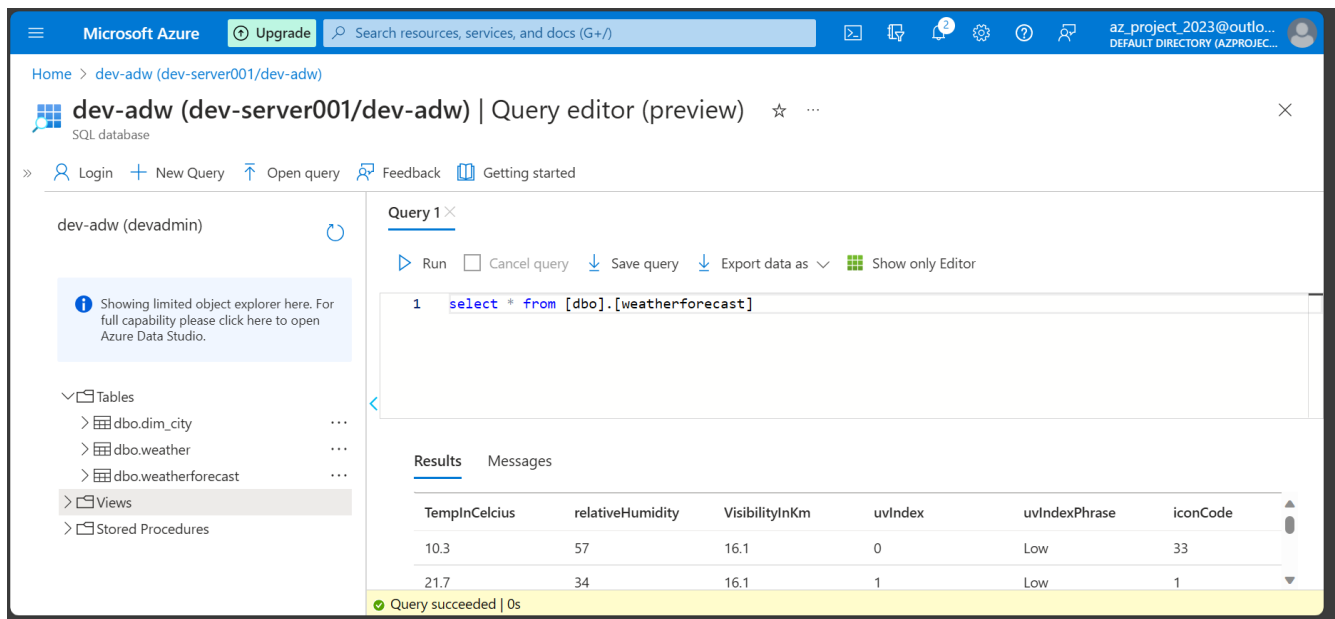
A.1: Containers to store raw and processed data in storage account



A.2: "WeatherForecast" table in Parquet form



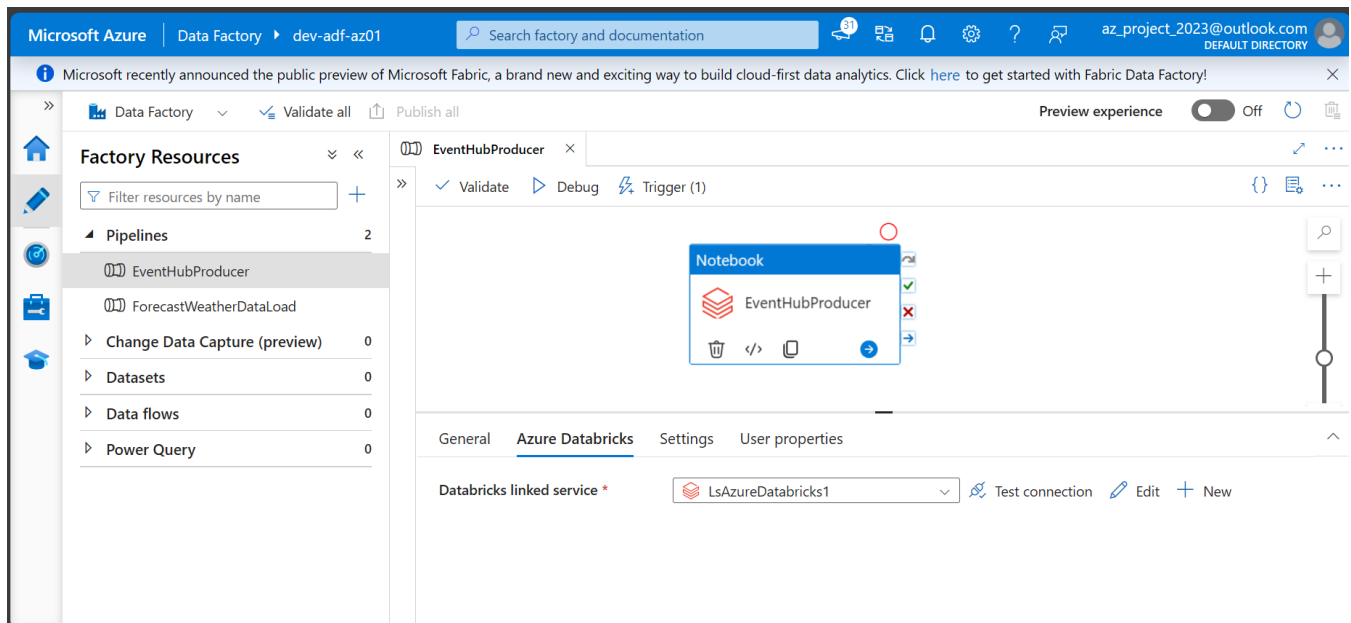
A.3 : Final tables are stored in SQL Server database.



The screenshot shows the Microsoft Azure portal interface for the 'dev-adw (dev-server001/dev-adw)' SQL database. The 'Query editor (preview)' is open, displaying a query: `select * from [dbo].[weatherforecast]`. The query has been executed successfully, as indicated by the 'Query succeeded | 0s' message. The results are shown in a table with the following columns: TempInCelcius, relativeHumidity, VisibilityInKm, uvIndex, uvIndexPhrase, and iconCode. The results table contains two rows of data.

TempInCelcius	relativeHumidity	VisibilityInKm	uvIndex	uvIndexPhrase	iconCode
10.3	57	16.1	0	Low	33
21.7	34	16.1	1	Low	1

A.4 : “EventHubProducer” pipeline to automate real-time weather data collection.



The screenshot shows the Microsoft Azure portal interface for the 'dev-adf-az01' Data Factory. The 'EventHubProducer' pipeline is selected, and the 'Notebook' activity is highlighted. The notebook activity is configured to run the 'EventHubProducer' script. The 'Databricks linked service' is set to 'LsAzureDatabricks1'. The pipeline is currently in the 'Validate' state.

A.5 : “ForecastWEatherDataLoad” pipeline to automate collection and processing of forecast data hourly

The screenshot displays the Microsoft Azure Data Factory (ADF) console. The left-hand navigation pane shows the 'Factory Resources' section with a list of items: Pipelines (2), EventHubProducer, ForecastWeatherDataLoad (selected), Change Data Capture (preview) (0), Datasets (0), Data flows (0), and Power Query (0). The main workspace shows the 'ForecastWeatherDataLoad' pipeline. At the top, there are buttons for 'Validate', 'Debug', and 'Trigger (1)'. A 'Notebook' icon is visible, labeled 'ForecastWeatherDataLoadMaster'. Below the pipeline name, there are tabs for 'General', 'Azure Databricks', 'Settings', and 'User properties'. The 'Azure Databricks' tab is active, showing a 'Databricks linked service' dropdown set to 'LsAzureDatabricks1', with options to 'Test connection', 'Edit', or 'New'.

A.6 : Stream Analytics job to process the streaming data coming from Event Hubs.

The screenshot shows the Microsoft Azure Stream Analytics console. The top navigation bar includes 'Home > Stream Analytics jobs > capture_stream_job1'. The main title is 'capture_stream_job1 | No-code editor (preview)'. Below the title, there are tabs for 'Data preview', 'Authoring errors', 'Runtime logs', and 'Metrics'. The 'Data preview' tab is active, showing a visual pipeline diagram. The pipeline consists of four stages: 1. 'Event Hub' with input 'weatherdata-eventhub'. 2. 'Expand' with output 'weather'. 3. 'Manage fields' with mappings: 'id → City Id', 'temp → Temp', and 'feels_like → Temp_feels_like'. 4. 'SQL' with output '[dbo].[weather]'. The diagram shows the flow from the Event Hub to the Expand stage, then to the Manage fields stage, and finally to the SQL stage.

B. Future Enhancements

The project's architecture allows for future enhancements such as:

- Integration with additional weather data sources.
- Implementation of machine learning models for more accurate forecasts.
- Extending the dashboard's capabilities for historical weather analysis.

C. References

- [Weather API - OpenWeatherMap](#)
- [azure-maps - Microsoft Azure](#)