Nikita H

Nikita-H

Nikita-H07

✉ nikita.h.0745@gmail.com

⛅ Real-Time Weather Monitoring Dashboard with Alerts

Project Overview

The **Real-Time Weather Monitoring Dashboard with Alerts** is a full-stack application designed to collect, display, and monitor weather data in real-time with aditional alert system to alert the user(e.g., high temperature, strong winds). It features dynamic visualizations, user authentication, alert mechanisms, and seamless integration with cloud infrastructure. This project showcases the practical use of cloud computing, third-party APIs, and data analytics for a responsive and scalable weather monitoring solution.

Key Features

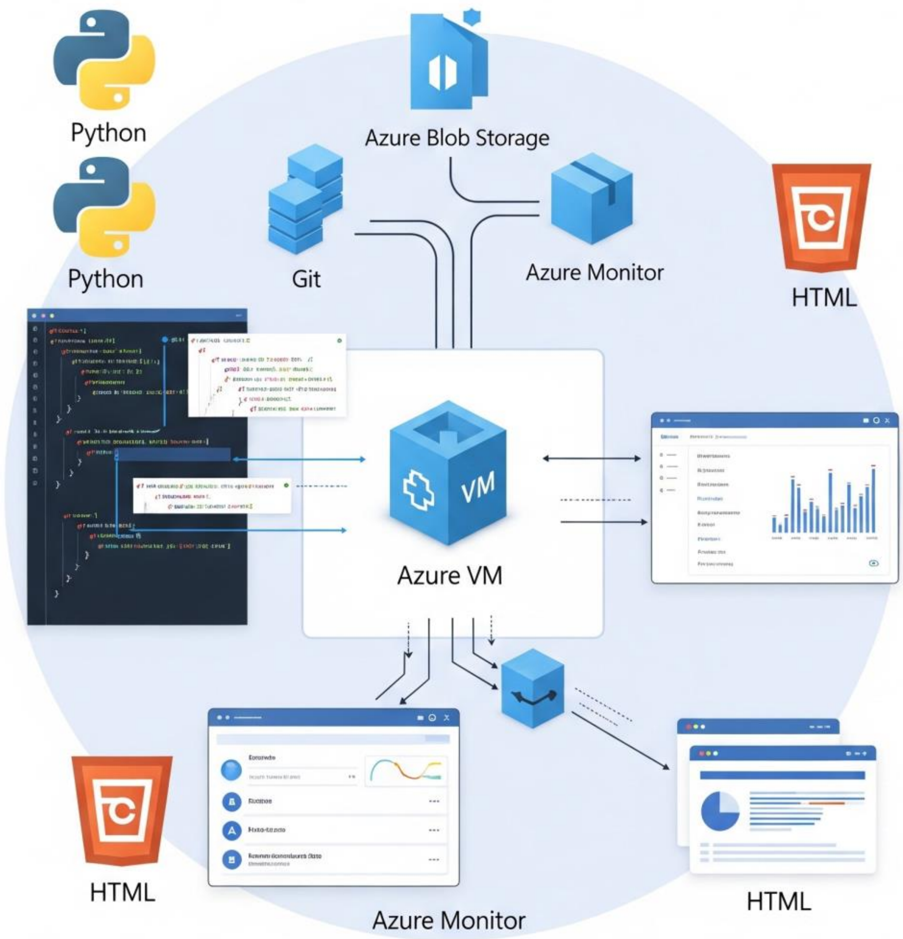
- **Real-time Weather Data:** Fetches live data using the OpenWeatherMap API.
- **User Authentication:** Secure login to access dashboard features.
- **Data Visualization:** Displays weather metrics like temperature, humidity, and wind speed using interactive charts.
- **Alert System:** Automatically triggers alerts when extreme or abnormal weather conditions are detected.
- **Cloud Deployment:** Hosted on a Microsoft Azure Virtual Machine with Azure Monitor enabled for performance tracking and logging.

Technologies Used

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** Python (Flask).
- **APIs:** OpenWeatherMap.
- **Cloud:** Microsoft Azure (Virtual Machine, Azure Monitor, Blob Storage).

Modules and Functionality

1. **Data Collection:** Periodically fetches live weather data via OpenWeatherMap API.
2. **Dashboard UI:** Charts dynamically update using Chart.js to reflect real-time changes.
3. **User Authentication:** Basic login system ensures secure access to the dashboard.
4. **Alerts:** Triggers visual and/or email alerts for values that exceed defined thresholds (e.g., high temperature, strong winds).
5. **Deployment:** Flask app is deployed on an Azure Virtual Machine with system monitoring via Azure Monitor.

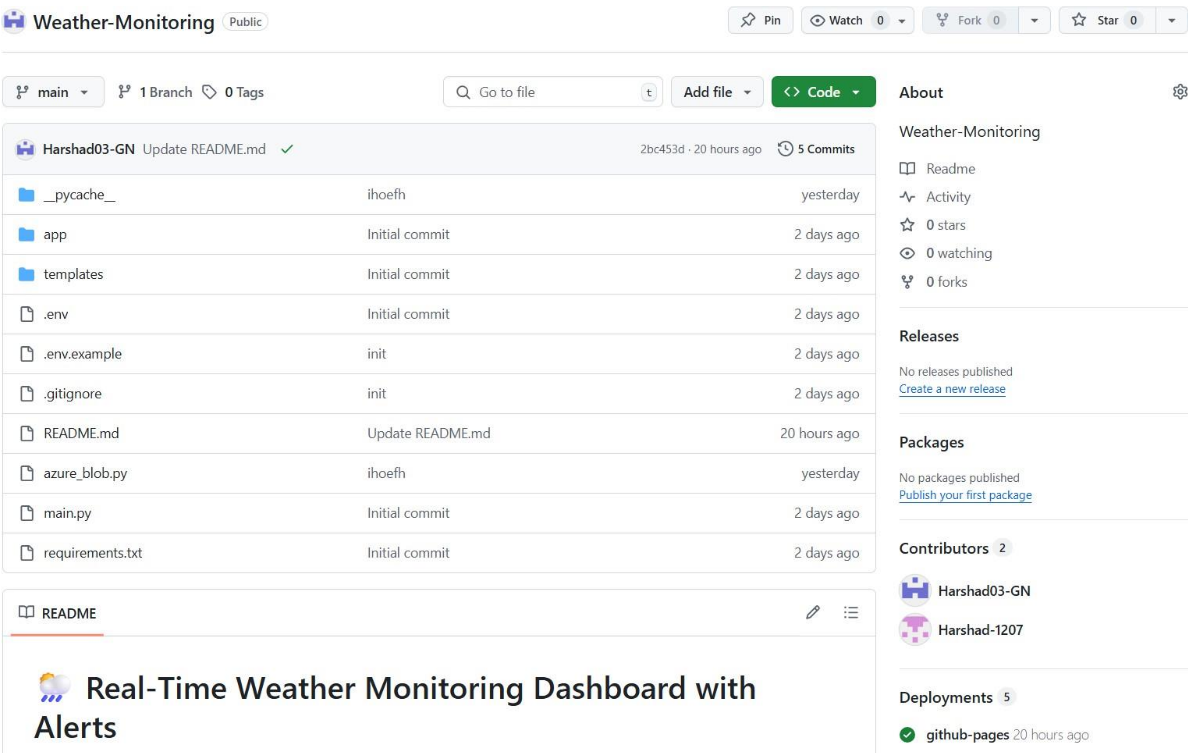

Python
Python
Azure Blob Storage
Git
Azure Monitor
HTML
VM
Azure VM
HTML
Azure Monitor
HTML

Project Timeline

Phase	Task	Duration
Phase 1	API Research and Planning	2 days
Phase 2	Frontend UI Development	3 days
Phase 3	Flask Backend Setup	2 days
Phase 4	Azure VM + Monitor Setup	2 days
Phase 5	Testing and Final Report	1 day

Challenges and Learnings

- Learned cloud deployment using Microsoft Azure.
- Faced API rate limit challenges and solved using caching.
- Understood Azure Monitor's diagnostic features.
- Improved frontend chart integration with asynchronous fetch.



Harshad03-GN Update README.md ✓ 2bc453d · 20 hours ago 5 Commits

__pycache__	ihoefh	yesterday
app	Initial commit	2 days ago
templates	Initial commit	2 days ago
.env	Initial commit	2 days ago
.env.example	init	2 days ago
.gitignore	init	2 days ago
README.md	Update README.md	20 hours ago
azure_blob.py	ihoefh	yesterday
main.py	Initial commit	2 days ago
requirements.txt	Initial commit	2 days ago

README

🌦️ Real-Time Weather Monitoring Dashboard with Alerts

About

Weather-Monitoring

Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

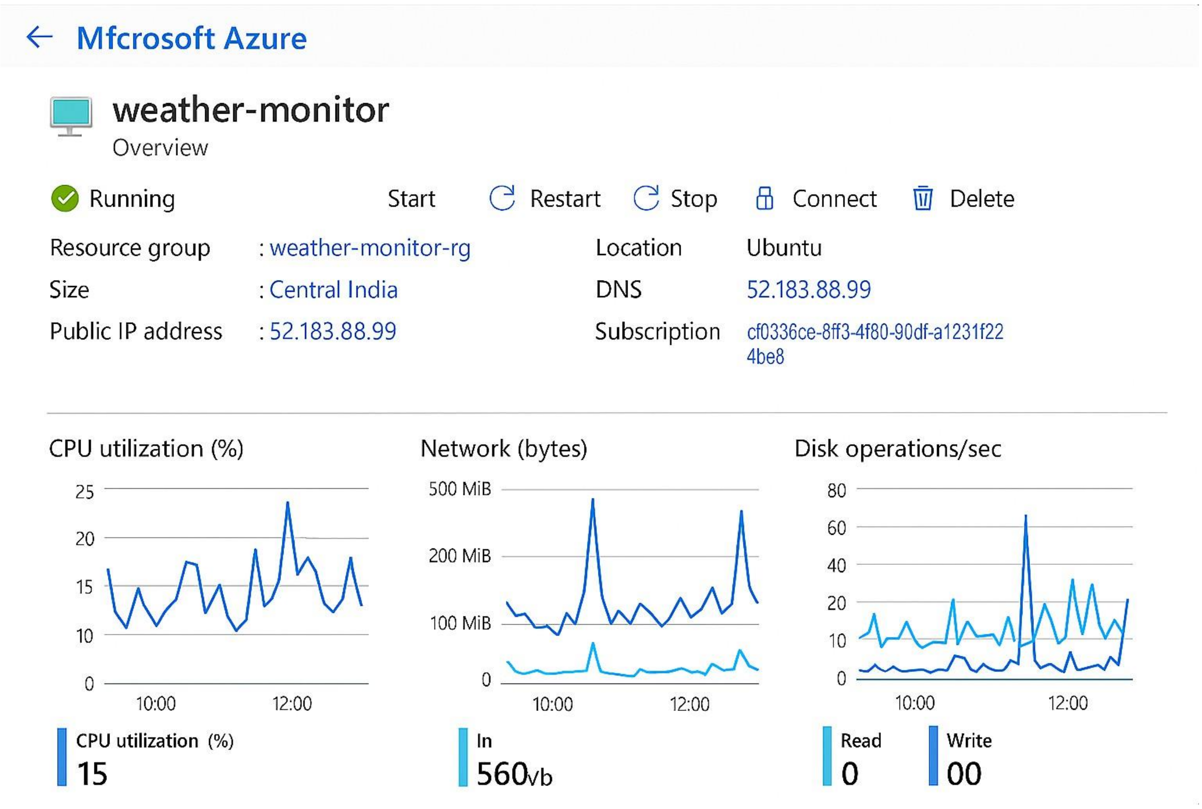
No packages published
Publish your first package

Contributors 2

Harshad03-GN
Harshad-1207

Deployments 5

✓ github-pages 20 hours ago



Mfcrosoft Azure

weather-monitor

Overview

Running

Resource group : weather-monitor-rg

Size : Central India

Public IP address : 52.183.88.99

Location Ubuntu

DNS 52.183.88.99

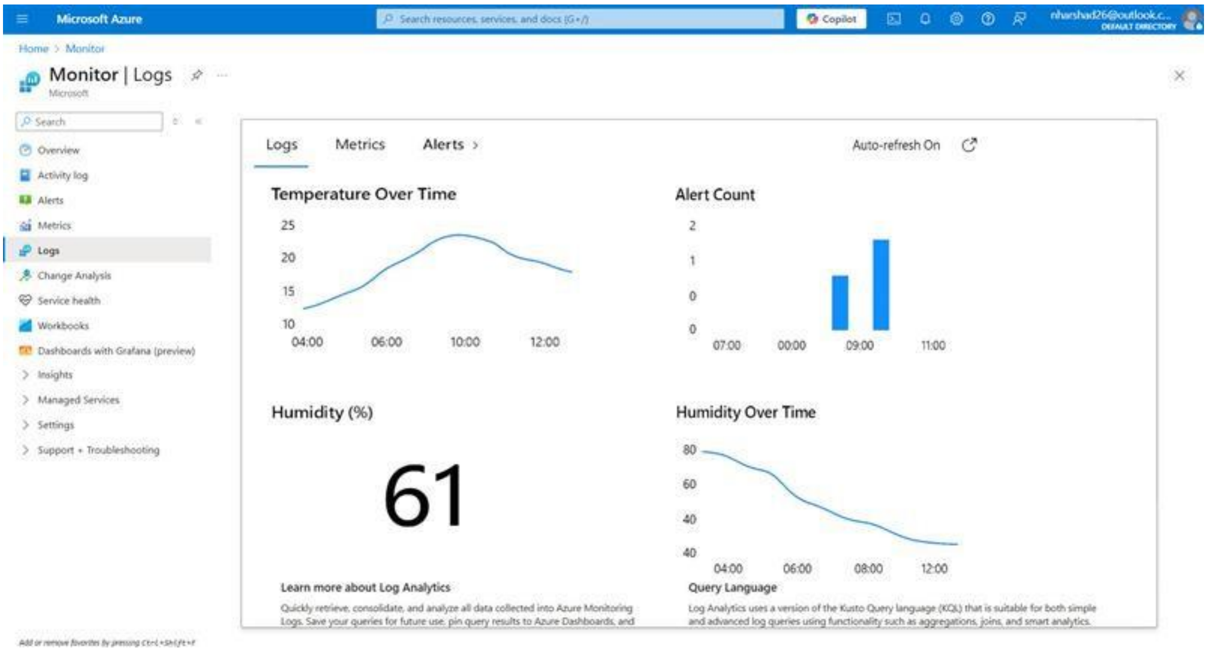
Subscription cf0336ce-8ff3-4f80-90df-a1231f22 4be8

CPU utilization (%) 15

In 560vb

Read 0

Write 00



Monitor | Logs

Microsoft

61

Learn more about Log Analytics

Quickly retrieve, consolidate, and analyze all data collected into Azure Monitoring Logs. Save your queries for future use, pin query results to Azure Dashboards, and

Query Language

Log Analytics uses a version of the Kusto Query language (KQL) that is suitable for both simple and advanced log queries using functionality such as aggregations, joins, and smart analytics.

Add or remove favorites by pressing Ctrl+Shift+F

Conclusion

This project provided a complete development and deployment experience in real-world cloud and DevOps practices. It showcased how weather data can be used in meaningful, user-friendly ways through real-time monitoring, dynamic visualization, and automated alert mechanisms. The alert system was a critical component, designed to detect abnormal weather conditions—such as extreme temperature, high humidity, or strong winds—and immediately notify users through visual warnings on the dashboard. This proactive approach ensured timely awareness and demonstrated how real-time data processing can enhance decision-making and safety in practical applications.

Acknowledgment

I would like to acknowledge that this cloud-based application project was conceptualized, developed, and deployed independently. The experience allowed me to explore various technologies, apply cloud computing principles, and gain hands-on practice with real-world development and DevOps workflows.

Connect: in Nikita-H07