

ĐẠI HỌC PHENIKAA
TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA

BÁO CÁO TỔNG QUAN

Ứng dụng phân tán

(Golang) influxdb - scalable datastore for metrics, events, and real-time analytics

((Golang) influxdb - kho dữ liệu có thể mở rộng cho số liệu, sự kiện và phân tích thời gian thực)

22010012 Trần Bá Tài 22010012@st.phenikaa-uni.edu.vn

Giảng viên hướng dẫn: ThS. Nguyễn Thành Trung
Khoa: Hệ thống thông tin

HÀ NỘI, 06/2025

Lời cam kết

Họ và tên nhóm sinh viên:

- Trần Bá Tài

Điện thoại liên lạc: 0941857885

Email: 22010012@st.phenikaa-uni.edu.vn

Lớp: K16 CNTT3

Hệ đào tạo: Cử nhân

Tôi cam kết Bài tập lớn (BTL) là công trình nghiên cứu của bản thân tôi. Các kết quả nêu trong BTL là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong BTL – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 05 tháng 06 năm 2025

Tác giả

Tài

Trần Bá Tài

Họ và tên sinh viên

Tóm tắt

Trong thời đại số hóa hiện nay, việc giám sát và theo dõi dữ liệu môi trường như nhiệt độ, độ ẩm hay áp suất khí quyển ngày càng trở nên quan trọng, đặc biệt trong các lĩnh vực nông nghiệp, y tế, và quản lý đô thị. Mặc dù đã có nhiều hệ thống dự báo và hiển thị thời tiết, phần lớn đều là dịch vụ tổng quát, thiếu khả năng tùy biến và cảnh báo theo nhu cầu cá nhân. Một số hệ thống cho phép trích xuất dữ liệu nhưng không hỗ trợ lưu trữ dài hạn hay phân tích tùy chỉnh, trong khi các nền tảng IoT chuyên dụng thì lại phức tạp và tốn chi phí triển khai.

Để giải quyết vấn đề trên, em lựa chọn hướng tiếp cận sử dụng **InfluxDB – cơ sở dữ liệu chuỗi thời gian**, kết hợp với Python để thu thập, lưu trữ, cảnh báo và hiển thị dữ liệu thời tiết từ OpenWeatherMap API. Hướng này đơn giản, dễ triển khai, chi phí thấp và đặc biệt phù hợp với bài toán thu thập dữ liệu định kỳ.

Giải pháp của em bao gồm ba chức năng chính:

- (1) thu thập dữ liệu thời tiết theo thời gian thực và lưu vào InfluxDB,
- (2) tự động gửi cảnh báo qua email khi dữ liệu vượt ngưỡng nguy hiểm
- (3) hiển thị dữ liệu lịch sử thông qua biểu đồ trên giao diện web.

Bài tập lớn này đã giúp em xây dựng một hệ thống hoàn chỉnh từ thu thập dữ liệu, lưu trữ, xử lý đến hiển thị. Kết quả là một mô hình ứng dụng phân tán đơn giản nhưng hiệu quả, có thể mở rộng cho các mục đích giám sát môi trường thực tế.

Mục lục

Lời cam kết	ii
Tóm tắt	iii
Mục lục	iv
Chương 1 Giới thiệu đề tài	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp	2
1.4 Bố cục bài tập lớn	3
Chương 2 Tổng quan dự án đã lựa chọn	4
2.1 . Tổng quan về InfluxDB	4
2.2 . Mục đích sử dụng của InfluxDB.....	6
2.3 . Chức năng chính của InfluxDB	7
2.4 . Ứng dụng thực tế	8
2.5 . Cài đặt Influxdb	9
Chương 3 Phát triển và triển khai kỹ thuật mới.....	12
3.1 Mô hình hệ thống tổng quát.....	12
3.2 Thu thập và lưu trữ dữ liệu thời tiết.....	13
3.3 Cảnh báo thời tiết qua email	13
3.4 Xây dựng giao diện web	14
3.5 Tự động hóa và triển khai	15
Chương 4 Kết luận và hướng phát triển	16

4.1 Kết luận.....	16
4.2 Hướng phát triển	16
Tài liệu tham khảo	18

Chương 1 Giới thiệu đề tài

Trong bối cảnh biến đổi khí hậu và đô thị hóa nhanh chóng, việc theo dõi các chỉ số thời tiết như nhiệt độ, độ ẩm, áp suất, và tốc độ gió ngày càng trở nên quan trọng trong nhiều lĩnh vực, từ nông nghiệp đến sức khỏe cộng đồng. Nhu cầu xây dựng một hệ thống giám sát thời tiết tùy biến, có khả năng cảnh báo và lưu trữ dữ liệu một cách hiệu quả, là một bài toán thực tiễn.

Với **InfluxDB – kho dữ liệu có thể mở rộng cho số liệu, sự kiện và phân tích thời gian thực** em xin nghiên cứu đề tài "**Xây dựng hệ thống giám sát và cảnh báo thời tiết sử dụng InfluxDB và Python**" được thực hiện nhằm mục tiêu xây dựng một mô hình ứng dụng phân tán đơn giản, có khả năng:

- Thu thập dữ liệu thời tiết từ API công khai (OpenWeatherMap),
- Lưu trữ dữ liệu dạng chuỗi thời gian bằng InfluxDB,
- Gửi cảnh báo qua email nếu dữ liệu vượt ngưỡng nguy hiểm,
- Hiển thị dữ liệu lịch sử bằng biểu đồ trên giao diện web.

Đề tài sử dụng các công nghệ phù hợp với ứng dụng phân tán, như: giao tiếp mạng (HTTP API), lưu trữ dữ liệu phân tán dạng time-series (InfluxDB), gửi thông báo qua SMTP, và xây dựng web server đơn giản bằng Flask.

Mục tiêu của đề tài không chỉ là xây dựng một hệ thống hoạt động ổn định, mà còn giúp sinh viên hiểu rõ quy trình thiết kế ứng dụng phân tán thực tế: từ tích hợp API, xử lý dữ liệu, đến phản hồi người dùng qua giao diện trực quan và cảnh báo tự động.

1.1 Đặt vấn đề

Trong thời đại hiện nay, dữ liệu thời tiết đóng vai trò quan trọng trong nhiều lĩnh vực như nông nghiệp, vận tải, giáo dục và y tế. Việc theo dõi liên tục và cảnh báo kịp thời khi các chỉ số thời tiết vượt quá mức cho phép có thể giúp con người đưa ra quyết định sớm và phù hợp. Tuy nhiên, nhiều hệ thống hiện có thường đòi hỏi chi phí cao, thiếu tính tùy biến và không hỗ trợ lưu trữ lịch sử để phân tích lâu dài. Từ đó, đặt ra nhu cầu xây dựng một hệ thống giám sát thời tiết tự động, có thể lưu trữ, phân tích và cảnh báo hiệu quả với chi phí thấp, dễ triển khai.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu chính của đề tài là thiết kế và xây dựng một hệ thống giám sát thời tiết có khả năng:

- Thu thập dữ liệu thời tiết theo thời gian thực từ nguồn dữ liệu công khai.
- Lưu trữ dữ liệu thời gian thực vào cơ sở dữ liệu InfluxDB.
- Gửi cảnh báo tự động qua email nếu các chỉ số thời tiết vượt ngưỡng nguy hiểm.
- Hiển thị biểu đồ dữ liệu lịch sử trên giao diện web thân thiện.

Phạm vi của đề tài bao gồm:

- Tích hợp API của OpenWeatherMap để lấy dữ liệu thời tiết tại một vị trí cụ thể.
- Xử lý và lưu dữ liệu vào hệ quản trị cơ sở dữ liệu time-series InfluxDB.
- Cảnh báo email dựa trên ngưỡng nhiệt độ, độ ẩm và áp suất do người lập trình định sẵn.
- Giao diện web đơn giản dùng Flask để hiển thị dữ liệu.

Đề tài không bao gồm:

- Phân tích dự báo thời tiết tương lai.
- Xử lý dữ liệu thời tiết đa điểm hoặc mở rộng quy mô phân tán nhiều node.

1.3 Định hướng giải pháp

Đề tài lựa chọn hướng tiếp cận sử dụng mô hình ứng dụng phân tán đơn giản với các thành phần độc lập nhưng liên kết chặt chẽ:

- **Nguồn dữ liệu:** Lấy từ OpenWeatherMap API.
- **Bộ xử lý dữ liệu và lưu trữ:** Dùng Python để xử lý và InfluxDB để lưu trữ dữ liệu thời gian.
- **Cảnh báo:** Sử dụng yagmail để gửi email tự động khi vượt ngưỡng.
- **Giao diện:** Hiển thị biểu đồ với Flask và Chart.js để trực quan hóa dữ liệu lịch sử.

Các thành phần này kết hợp với nhau thông qua các giao thức mạng chuẩn (HTTP, SMTP), phù hợp với tiêu chuẩn của một hệ thống ứng dụng phân tán.

1.4 Bố cục bài tập lớn

Bài báo cáo được tổ chức thành các phần chính sau:

1. Giới thiệu đề tài

Trình bày lý do lựa chọn đề tài, tính thực tiễn và các vấn đề cần giải quyết.

2. Cơ sở lý thuyết và công nghệ

Giới thiệu về các công nghệ sử dụng trong đề tài như OpenWeatherMap API, InfluxDB, Flask, SMTP, Chart.js,...

3. Phân tích và thiết kế hệ thống

Mô tả kiến trúc tổng thể, sơ đồ luồng dữ liệu, mô hình thành phần, các chức năng chính và cách tương tác giữa các module.

4. Cài đặt và thử nghiệm

Trình bày quá trình cài đặt các thành phần, mã nguồn chính, các bước triển khai và kết quả thử nghiệm.

5. Đánh giá và kết luận

Tổng kết những gì đạt được, phân tích ưu/nhược điểm, và định hướng phát triển sau này.

Chương 2 Tổng quan dự án đã lựa chọn

(InfluxDB – kho dữ liệu có thể mở rộng cho số liệu, sự kiện và phân tích thời gian thực)

2.1 . Tổng quan về InfluxDB

1.1. Giới thiệu về InfluxDB

InfluxDB là một hệ quản trị cơ sở dữ liệu chuỗi thời gian (Time Series Database - TSDB) mã nguồn mở, được tối ưu hóa để lưu trữ, truy vấn và xử lý các dữ liệu biến đổi theo thời gian. Nó đặc biệt phù hợp với các ứng dụng yêu cầu ghi nhận và phân tích dữ liệu theo từng mốc thời gian như giám sát hệ thống, dữ liệu cảm biến IoT, dữ liệu giao dịch tài chính, log hệ thống và các phép đo trong khoa học.

InfluxDB là sản phẩm chủ lực của **InfluxData**, một công ty công nghệ có trụ sở tại San Francisco, Hoa Kỳ, chuyên cung cấp giải pháp dữ liệu thời gian thực. Hệ sinh thái InfluxData bao gồm:

- **Telegraf** – công cụ thu thập dữ liệu nhẹ, plugin-driven.
- **InfluxDB** – cơ sở dữ liệu lưu trữ dữ liệu thời gian.
- **Chronograf** – giao diện trực quan hóa và quản trị.
- **Kapacitor** – công cụ xử lý dữ liệu thời gian thực và cảnh báo.

Ngôn ngữ lập trình chủ đạo được sử dụng để phát triển InfluxDB là **Go (Golang)** – một ngôn ngữ nổi bật về khả năng đồng thời (concurrency), hiệu năng cao, gọn nhẹ và dễ triển khai trên nhiều nền tảng (Linux, Windows, macOS...).

InfluxDB hỗ trợ nhiều giao thức nhập dữ liệu như HTTP API, UDP, MQTT, và hỗ trợ các thư viện client phong phú cho Python, Java, Go, JavaScript, C#, v.v... giúp dễ dàng tích hợp vào bất kỳ hệ thống nào.

1.2. Quá trình phát triển

InfluxDB đã trải qua nhiều giai đoạn phát triển, từ phiên bản ban đầu với mục tiêu đơn giản là lưu trữ dữ liệu thời gian thực đến một nền tảng dữ liệu toàn diện trong phiên bản 2.x và 3.x.

Giai đoạn 1: InfluxDB 0.x đến 1.x (2013 – 2019)

- InfluxDB sử dụng InfluxQL làm ngôn ngữ truy vấn cốt lõi, có cú pháp tương tự SQL, giúp người dùng dễ dàng truy vấn dữ liệu thời gian. Dữ liệu được lưu trữ theo kiến trúc TSM (Time-Structured Merge Tree), tối ưu cho việc ghi nhanh và truy xuất theo mốc thời gian. Hệ thống hỗ trợ các tính năng cơ bản như tạo cơ sở dữ liệu, thiết lập chính sách lưu giữ (retention policy), truy vấn liên tục (continuous query), quản lý series và shard. Tuy nhiên, InfluxDB vẫn còn một số hạn chế như giao diện người dùng đơn giản, thiếu cơ chế phân quyền mạnh và chưa hỗ trợ tốt các thao tác xử lý dữ liệu phức tạp

Giai đoạn 2: InfluxDB 2.x (2019 – nay)

- Bên cạnh InfluxQL, InfluxDB còn giới thiệu ngôn ngữ Flux – một ngôn ngữ được xây dựng từ đầu nhằm hỗ trợ xử lý, biến đổi và phân tích dữ liệu chuỗi thời gian một cách linh hoạt và mạnh mẽ hơn. Giao diện người dùng (UI) của InfluxDB cũng đã được hoàn thiện với các tính năng quản lý, xây dựng dashboard, thiết lập cảnh báo (alert) và tích hợp nhiều nguồn dữ liệu khác nhau. InfluxDB Cloud là phiên bản SaaS chạy trên các nền tảng đám mây phổ biến như AWS, GCP và Azure, mang lại khả năng mở rộng linh hoạt và triển khai nhanh chóng. Với kiến trúc "Single Binary", toàn bộ hệ thống có thể vận hành chỉ trong một tệp tin thực thi duy nhất. Ngoài ra, hệ thống còn tích hợp sẵn các chức năng quản lý tổ chức, bucket, token và dashboard, giúp việc cấu hình và sử dụng trở nên đơn giản và hiệu quả hơn.

Giai đoạn 3: InfluxDB 3.x (2023 – nay)

- InfluxDB hiện đang chuyển sang kiến trúc lưu trữ mới dựa trên định dạng columnar Parquet, cho phép xử lý hiệu quả khối lượng dữ liệu cực lớn. Việc tích hợp với các công nghệ như Apache Arrow và DataFusion giúp nâng cao hiệu suất truy vấn và khả năng tương tác với các hệ thống phân tích hiện đại. Kiến trúc mới này cũng định hướng sử dụng trong các hệ thống Data Lake, hỗ trợ truy vấn song song dựa trên SQL bên cạnh ngôn ngữ Flux. Với những cải tiến này, InfluxDB hướng tới khả năng mở rộng quy mô mạnh mẽ, tích hợp sâu với các hệ thống AI/ML và mang lại tốc độ truy vấn vượt trội so với phiên bản 2.x trước đó.

1.3. Mô hình dữ liệu InfluxDB

Dữ liệu trong InfluxDB tuân theo mô hình dữ liệu phi quan hệ, không cần định nghĩa schema trước, và được ghi nhận bằng cú pháp **Line Protocol** với 4 thành phần chính:

1. Measurement (Đo lường)

- Tương tự như bảng (table) trong RDBMS.

- VD: sensor_data, cpu_usage, weather.

2. Tags (Nhãn)

- Metadata để nhóm và lọc dữ liệu.
- Là khóa-giá trị, **được index** để tối ưu truy vấn.
- VD: location=Hanoi, device=iot123.

3. Fields (Trường dữ liệu)

- Giá trị đo đạc: số, chuỗi, boolean.
- **Không được index**, nhưng bắt buộc mỗi điểm phải có ít nhất 1 field.
- VD: temperature=30.5, status="OK".

4. Timestamp (Thời gian)

- Dấu thời gian chính xác đến nanosecond.
- Là khóa chính xác định duy nhất một điểm dữ liệu.

Series: một Series là tổ hợp giữa Measurement + Tag Set + Field Key.

- VD: weather,location=Hanoi temperature.

Series Cardinality: số lượng series duy nhất. Đây là yếu tố ảnh hưởng mạnh tới hiệu suất hệ thống – cần tối ưu tránh số lượng series quá lớn (cardinality explosion).

2.2 . Mục đích sử dụng của InfluxDB

InfluxDB được thiết kế chuyên biệt để xử lý **dữ liệu thời gian thực**, với các mục tiêu chính:

- **Thu thập dữ liệu liên tục:** từ cảm biến, log hệ thống, API, hoặc file.
- **Truy vấn hiệu quả:** thực hiện các phép toán theo thời gian như trung bình, tổng, đếm, max, min, v.v...
- **Lưu trữ tối ưu:** sử dụng cơ chế nén dữ liệu, lưu trữ phân đoạn (shard) giúp tiết kiệm tài nguyên.
- **Tích hợp với các công cụ phân tích, dashboard** như Grafana, hoặc gửi alert qua email, Slack.

2.3 . Chức năng chính của InfluxDB

3.1. Cấu trúc dữ liệu chuyên biệt

Dữ liệu trong InfluxDB được tối ưu hóa đặc biệt cho các truy vấn theo trục thời gian, giúp tăng tốc độ truy xuất và phân tích theo mốc thời gian cụ thể. Một ưu điểm nổi bật của hệ thống là không yêu cầu người dùng phải định nghĩa schema trước, mang lại sự linh hoạt cao trong việc thu thập và lưu trữ dữ liệu. Nhờ vào kiến trúc thiết kế chuyên biệt, InfluxDB có khả năng ghi nhận hàng triệu điểm dữ liệu mỗi giây, đáp ứng tốt nhu cầu của các hệ thống giám sát, IoT và các ứng dụng thời gian thực quy mô lớn.

3.2. Ngôn ngữ truy vấn

InfluxDB hỗ trợ hai ngôn ngữ truy vấn chính: InfluxQL và Flux. InfluxQL có cú pháp gần giống SQL, giúp người mới dễ tiếp cận và học nhanh chóng khi làm việc với dữ liệu chuỗi thời gian. Trong khi đó, Flux là một ngôn ngữ truy vấn mạnh mẽ hơn, được thiết kế để xử lý dòng dữ liệu linh hoạt, hỗ trợ các thao tác phức tạp như điều kiện logic, join, pivot, map, reduce và truy vấn dữ liệu từ nhiều bucket cùng lúc. Sự kết hợp này giúp InfluxDB vừa thân thiện với người dùng mới, vừa đáp ứng được nhu cầu phân tích nâng cao trong các hệ thống dữ liệu hiện đại.

3.3. Quản lý dữ liệu

InfluxDB cung cấp các cơ chế quản lý dữ liệu theo thời gian giúp tối ưu lưu trữ và hiệu suất. Retention Policy cho phép định nghĩa thời gian dữ liệu được giữ lại trong hệ thống, tự động xóa dữ liệu cũ sau khoảng thời gian nhất định. Continuous Query là các truy vấn chạy định kỳ nhằm tự động tổng hợp hoặc chuyển đổi dữ liệu, giảm tải cho truy vấn thời gian thực. Kết hợp với Downsampling, InfluxDB hỗ trợ lưu trữ dữ liệu thô trong thời gian ngắn để phân tích chi tiết, đồng thời lưu giữ các bản tổng hợp dữ liệu trong thời gian dài hơn, giúp cân bằng giữa chi phí lưu trữ và khả năng phân tích lịch sử.

3.4. Khả năng tích hợp

InfluxDB hỗ trợ hệ thống plugin mạnh mẽ cho phép gửi cảnh báo (alert) và kích hoạt các hành động xử lý theo thời gian thực. Nhờ khả năng tích hợp tốt với các hệ thống giám sát phổ biến như Prometheus, Grafana, ELK, v.v., InfluxDB dễ dàng trở thành một phần trong kiến trúc quan sát tổng thể của doanh nghiệp. Các cảnh báo có thể được cấu hình để gửi qua nhiều kênh như Slack, Email, Webhook và các nền tảng khác, giúp người dùng kịp thời phát hiện và phản ứng với các sự kiện quan trọng trong hệ thống.

2.4 . Ứng dụng thực tế

4.1. Giám sát hệ thống CNTT

- Thu thập dữ liệu từ máy chủ, ứng dụng, thiết bị mạng.
- Theo dõi CPU, RAM, disk, network theo thời gian thực.
- Kết hợp Grafana để vẽ biểu đồ cảnh báo.

4.2. Internet of Things (IoT)

- Lưu trữ dữ liệu cảm biến: nhiệt độ, độ ẩm, ánh sáng, CO2...
- Xây dựng dashboard phân tích tại nhà máy, nông trại thông minh.
- Phân tích xu hướng, tự động cảnh báo sự cố.

4.3. Tài chính & Giao dịch

- Phân tích dòng tiền, số lượng giao dịch, giá cổ phiếu.
- Phân tích dữ liệu ticker, biểu đồ nến thời gian thực.
- Dự đoán xu hướng đầu tư dựa trên lịch sử dữ liệu.

4.4. Y tế

- Theo dõi nhịp tim, huyết áp, ECG trong ICU.
- Lưu trữ dữ liệu từ thiết bị wearable (Apple Watch, Fitbit).
- Phân tích dữ liệu nghiên cứu lâm sàng dài hạn.

4.5. Nghiên cứu và khoa học

- Ghi nhận dữ liệu cảm biến từ trạm khí tượng, môi trường.
- Ghi nhận thay đổi địa chấn, mức nước biển, thời tiết...

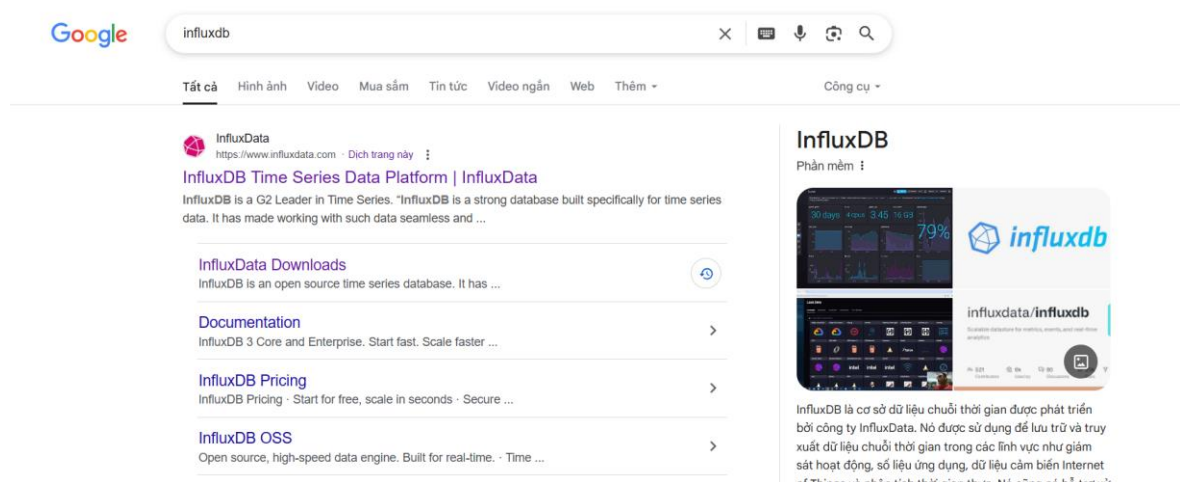
Kết luận

InfluxDB không chỉ là một cơ sở dữ liệu chuỗi thời gian, mà là một nền tảng hoàn chỉnh để xây dựng các hệ thống theo dõi và phân tích dữ liệu thời gian thực với hiệu suất cao, chi phí thấp và tích hợp linh hoạt. Với tốc độ phát triển nhanh và cộng đồng lớn mạnh, InfluxDB

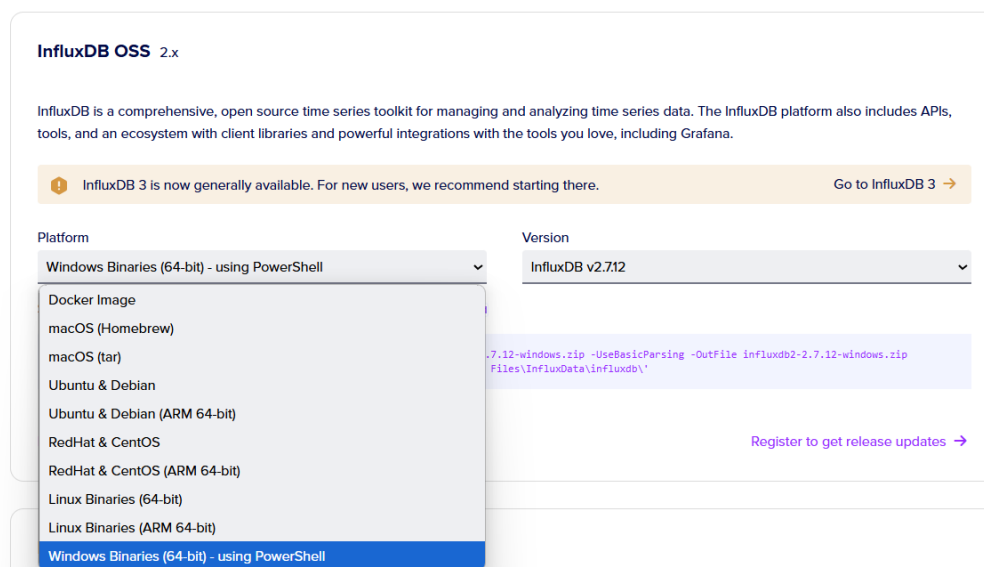
đang trở thành lựa chọn hàng đầu cho các ứng dụng cần xử lý dữ liệu thời gian thực trong nhiều lĩnh vực công nghệ, công nghiệp, y tế, tài chính và nghiên cứu.

2.5 . Cài đặt Influxdb

Bước 1 : Truy cập influxdata.com/downloads/



Bước 2 : Sau đó kéo xuống bên dưới đến InfluxDB OSS 2.x và chọn platform tương ứng với máy của bạn và Version mới nhất



Sau đó với Windows copy đoạn mã dán vào PowerShell của máy bạn để tải :

```
wget https://download.influxdata.com/influxdb/releases/v2.7.12/influxdb2-2.7.12-windows.zip -UseBasicParsing -OutFile influxdb2-2.7.12-windows.zip
Expand-Archive .\influxdb2-2.7.12-windows.zip -DestinationPath 'C:\Program Files\InfluxData\influxdb\'
```

Nếu máy bạn chưa cài wget có 2 cách :

- Cài wget for windows :

Wget for Windows

Wget: retrieve files from the WWW

Version

1.11.4

Description

GNU Wget is a free network utility to retrieve files from the World Wide Web using HTTP and FTP, the two most widely used Internet protocols.

The recursive retrieval of HTML pages, as well as FTP sites is supported -- you can use Wget to make mirrors of archives and home pages.

Wget works exceedingly well on slow or unstable connections, keeping getting the document until it is fully retrieved. Re-getting and recursive mirroring of directories are available when retrieving via FTP. Both HTTP and FTP retrievals can be time-stamped with the version if it has.

Wget supports proxy servers, which can lighten the network load, speed up retrieval and provide access behind firewalls. If you want to use a proxy, compile wget with support for socks.

Most of the features are configurable, either through command-line options, or via initialization file .wgetrc. Wget allows you to

Homepage

<http://www.gnu.org/software/wget>

Sources: <http://ftp.gnu.org/gnu/wget>

Download

- Chạy câu lệnh tương ứng với Windows:

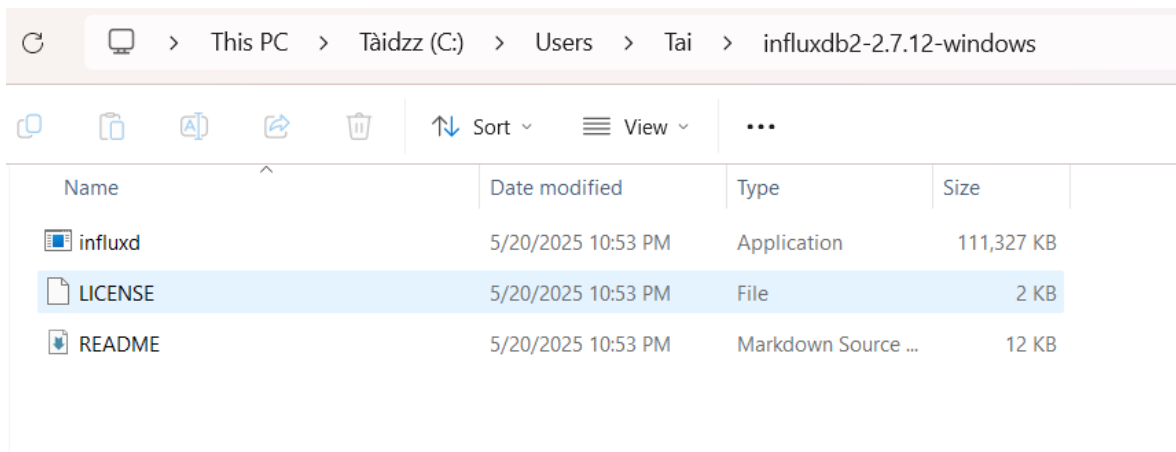
```
Invoke-WebRequest -Uri "https://download.influxdata.com/influxdb/releases/v2.7.12/influxdb2-2.7.12-windows.zip" -OutFile "influxdb2-2.7.12-windows.zip"
```

- Hoặc dùng curl (nếu có) :

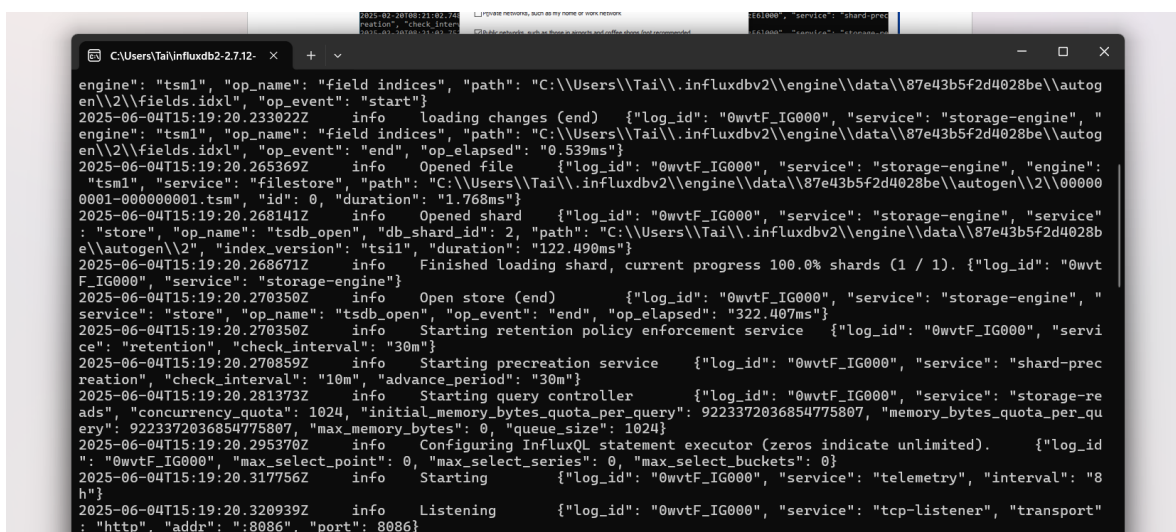
```
curl -L -o influxdb2-2.7.12-windows.zip https://download.influxdata.com/influxdb/releases/v2.7.12/influxdb2-2.7.12-windows.zip
```

Bước 3 : Sau khi tải xong bạn sẽ có 1 file zip : influxdb2-2.7.12-windows.zip

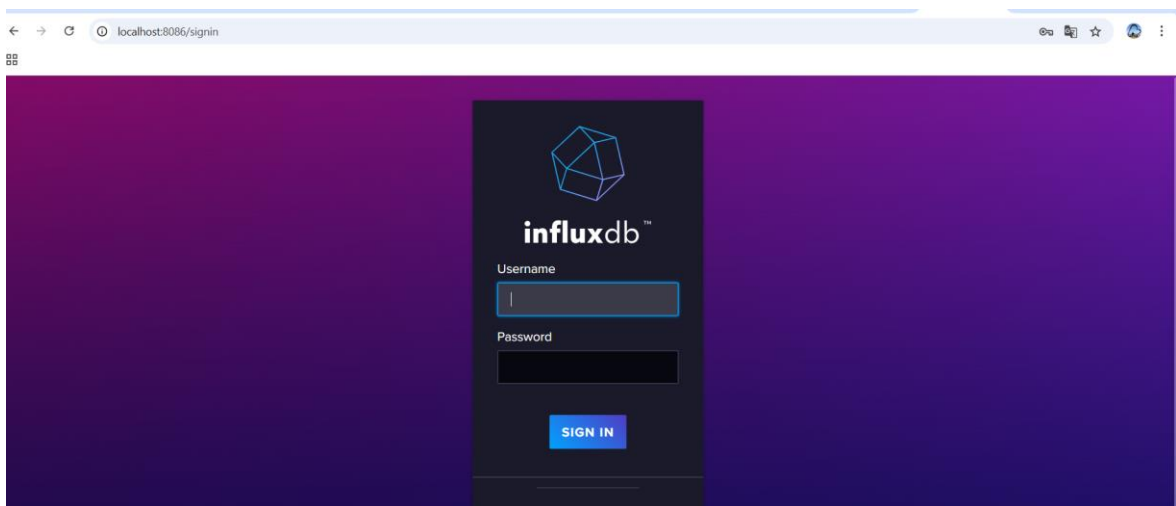
Bạn cần giải nén nó và tìm đến file influxd.exe để cài đặt influxdb :



Sau khi cài xong sẽ hiện lên địa chỉ để ta truy cập vào mặc định là Port 8086



Để mở serve ta sẽ tìm kiếm trên trình duyệt : localhost:8086



Bước 4 : Đăng ký tài khoản và sử dụng influxdb

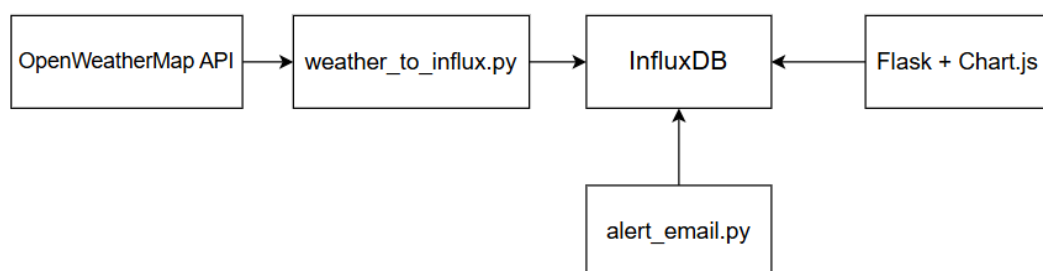
Chương 3 Phát triển và triển khai kỹ thuật mới

3.1 Mô hình hệ thống tổng quát

Hệ thống được xây dựng gồm các thành phần chính:

- **Nguồn dữ liệu:** API của OpenWeatherMap cung cấp dữ liệu thời tiết theo thời gian thực.
- **Bộ thu thập dữ liệu (Collector):** Một đoạn script Python định kỳ gọi API và ghi dữ liệu thời tiết vào cơ sở dữ liệu.
- **Kho lưu trữ dữ liệu thời gian thực:** InfluxDB đóng vai trò lưu trữ dữ liệu với các thông số như: nhiệt độ, độ ẩm, áp suất, tốc độ gió,...
- **Hệ thống cảnh báo:** Một module kiểm tra ngưỡng thời tiết và gửi email cảnh báo nếu phát hiện điều kiện bất thường.
- **Giao diện Web:** Ứng dụng Flask kết hợp với Chart.js hiển thị biểu đồ dữ liệu thời tiết lịch sử cho người dùng.

Sơ đồ hoạt động:



3.2 Thu thập và lưu trữ dữ liệu thời tiết

Một script Python (weather_to_influx.py) được lập trình để:

- Gửi request đến OpenWeatherMap API để lấy dữ liệu tại thành phố Hà Nội.
- Trích xuất các thông số cần thiết: nhiệt độ, độ ẩm, tốc độ gió,...
- Ghi dữ liệu vào InfluxDB với measurement = "weather", timestamp được hệ thống tự động thêm.

Dữ liệu được ghi định kỳ mỗi vài phút bằng cách chạy script liên tục, hoặc tích hợp thêm scheduler để tự động hóa.

```
weather_to_influx.py > ...
13 def fetch_weather():
23
24     wind_speed = data["wind"]["speed"]
25     return temp, humidity, wind_speed
26
27 def write_to_influx(temp, humidity, wind_speed):
28     with InfluxDBClient(url=INFLUXDB_URL, token=INFLUXDB_TOKEN, org=ORG) as client:
29         write_api = client.write_api()
30         point = (
31             Point("weather")
32             .field("temperature", temp)
33             .field("humidity", humidity)
34             .field("wind_speed", wind_speed)
35             .time(time.time_ns(), WritePrecision.NS)
36         )
37         write_api.write(bucket=BUCKET, record=point)
38         print(f"✅ {datetime.now()} - Ghi: {temp}°C | {humidity}% | {wind_speed} m/s")
39
40 if __name__ == "__main__":
41     while True:
42         try:
43             temp, humidity, wind_speed = fetch_weather()
44             write_to_influx(temp, humidity, wind_speed)
45         except Exception as e:
46             print(f"❌ {datetime.now()} - Lỗi: {e}")
47             time.sleep(300)
48
```

3.3 Cảnh báo thời tiết qua email

Một module Python khác (alert_email.py) thực hiện truy vấn dữ liệu gần nhất từ InfluxDB:

- So sánh với các ngưỡng cảnh báo định sẵn (ví dụ: nhiệt độ > 30°C, độ ẩm > 80%, gió > 10 m/s).
- Nếu vượt ngưỡng, hệ thống gửi email cảnh báo đến người dùng sử dụng thư viện yagmail và SMTP Gmail.
- Module được thiết lập chạy tự động bằng cách sử dụng schedule hoặc threading.Timer.

```

alert_email.py > check_and_send_alert
39 def check_and_send_alert(weather):
44     alerts = []
45     if temp > 35:
46         alerts.append(f"Nhiệt độ cao: {temp}°C")
47     if humidity > 80:
48         alerts.append(f"Độ ẩm cao: {humidity}%")
49     if wind > 10:
50         alerts.append(f"Gió mạnh: {wind} m/s")
51
52     if alerts:
53         subject = "⚠️ CẢNH BÁO THỜI TIẾT"
54         body = "\n".join(alerts)
55         yag = yagmail.SMTP(EMAIL_SENDER, EMAIL_PASSWORD)
56         yag.send(to=EMAIL_RECEIVER, subject=subject, contents=body)
57         print(f"\n✅ {datetime.now()} - Đã gửi email cảnh báo:\n{body}")
58     else:
59         print(f"\n✅ {datetime.now()} - Không có điều kiện cảnh báo.")
60
61 if __name__ == "__main__":
62     while True:
63         try:
64             data = get_latest_weather()
65             check_and_send_alert(data)
66         except Exception as e:
67             print(f"\n❌ {datetime.now()} - Lỗi: {e}")
68             time.sleep(100)

```

3.4 Xây dựng giao diện web

Ứng dụng web sử dụng Flask để cung cấp:

- Trang chủ hiển thị biểu đồ nhiệt độ, độ ẩm, áp suất theo thời gian.
- Truy vấn InfluxDB bằng API và vẽ biểu đồ bằng thư viện Chart.js phía frontend.
- Giao diện trực quan giúp người dùng theo dõi lịch sử thời tiết một cách dễ dàng.

```

template > index.html > html > body > script > drawChart
2 <html lang="en">
3 </html>
4 <body>
5 <h2>🌤️ Biểu đồ thời tiết 1 giờ gần nhất</h2>
6 <canvas id="weatherChart" width="800" height="400"></canvas>
7
8 <script>
9     async function drawChart() {
10         const res = await fetch('/api/weather-data');
11         const data = await res.json();
12
13         const ctx = document.getElementById('weatherChart').getContext('2d');
14         new Chart(ctx, {
15             type: 'line',
16             data: {
17                 labels: data.timestamps,
18                 datasets: [
19                     {
20                         label: 'Nhiệt độ (°C)',
21                         data: data.temperature,
22                         borderColor: 'red',
23                         fill: false
24                     },
25                     {
26                         label: 'Độ ẩm (%)',
27                         data: data.humidity,
28                         borderColor: 'blue',
29                         fill: false
30                     }
31                 ]
32             }
33         });
34     }
35     drawChart();
36 </script>

```

3.5 Tự động hóa và triển khai

Hệ thống được cấu hình sao cho toàn bộ quá trình vận hành có thể tự động:

- Thu thập và ghi dữ liệu: chạy định kỳ.
- Kiểm tra và gửi cảnh báo: chạy nền tự động.
- Giao diện web: chạy qua Flask server, có thể truy cập từ localhost.

Chương 4 Kết luận và hướng phát triển

4.1 Kết luận

Qua quá trình nghiên cứu **InfluxDB – kho dữ liệu có thể mở rộng cho số liệu, sự kiện và phân tích thời gian thực** và thực hiện đề tài “Xây dựng hệ thống giám sát và cảnh báo thời tiết thời gian thực sử dụng InfluxDB”, em đã xây dựng thành công một hệ thống hoàn chỉnh bao gồm các chức năng chính: thu thập dữ liệu thời tiết thời gian thực từ API OpenWeatherMap, lưu trữ dữ liệu hiệu quả bằng cơ sở dữ liệu thời gian InfluxDB, tự động phát hiện điều kiện thời tiết bất thường và gửi cảnh báo qua email, cũng như cung cấp giao diện web trực quan hiển thị lịch sử các thông số thời tiết.

Hệ thống đã vận hành ổn định trong môi trường thử nghiệm cục bộ, cho phép người dùng theo dõi và nhận thông tin về thời tiết một cách nhanh chóng và chính xác. Các công nghệ mã nguồn mở như InfluxDB, Flask và Python đã được kết hợp hiệu quả để xây dựng một giải pháp nhẹ, linh hoạt nhưng vẫn đảm bảo các yêu cầu về thời gian thực và khả năng tự động hóa.

Bài tập lớn này không chỉ giúp sinh viên hiểu rõ hơn về việc ứng dụng cơ sở dữ liệu thời gian thực trong các bài toán thực tiễn, mà còn rèn luyện kỹ năng tích hợp nhiều công nghệ khác nhau trong một hệ thống phân tán nhỏ gọn.

Em xin gửi lời cảm ơn chân thành đến **Thầy Thạc sĩ Nguyễn Thành Trung**, giảng viên hướng dẫn môn *Ứng dụng phân tán*, người đã tận tình truyền đạt kiến thức, định hướng và hỗ trợ em trong suốt quá trình thực hiện bài tập lớn này. Những góp ý, các tài liệu tham khảo và sự hỗ trợ chuyên môn từ Thầy đã giúp em hoàn thiện hệ thống một cách hiệu quả và học hỏi thêm nhiều kỹ năng thực tiễn.

4.2 Hướng phát triển

Trong tương lai, hệ thống có thể được mở rộng theo các hướng sau:

- **Mở rộng vùng địa lý:** Cho phép người dùng lựa chọn nhiều thành phố hoặc khu vực để theo dõi cùng lúc.
- **Giao diện người dùng nâng cao:** Phát triển dashboard trực quan hơn với nhiều biểu đồ, phân tích thống kê và biểu diễn dữ liệu đa dạng.

- **Tích hợp cảnh báo đa kênh:** Ngoài email, hệ thống có thể gửi cảnh báo qua Telegram, Zalo hoặc SMS để nâng cao độ phản hồi.
- **Triển khai đám mây:** Đưa toàn bộ hệ thống lên nền tảng đám mây như AWS hoặc Heroku để đảm bảo hoạt động 24/7 và hỗ trợ mở rộng quy mô.
- **Phân quyền người dùng và cá nhân hóa:** Cho phép người dùng đăng ký tài khoản, tùy chỉnh ngưỡng cảnh báo và nhận thông tin cá nhân hóa.
- **Tạo ra được ứng dụng và thích ứng trên mọi thiết bị**

Với nền tảng đã xây dựng, hệ thống hoàn toàn có thể trở thành một ứng dụng thực tế trong các lĩnh vực như nông nghiệp thông minh, cảnh báo thiên tai, hoặc các hệ thống IoT thu thập dữ liệu môi trường tự động.

Tài liệu tham khảo

<https://github.com/influxdata/influxdb>

<https://www.influxdata.com/>

<https://en.wikipedia.org/wiki/InfluxDB>

<https://openweathermap.org/>

<https://docs.influxhq.com/>

https://www.youtube.com/watch?v=XloH_0G2IzA

<https://docs.influxdata.com/influxdb/v2/get-started/setup/>

<https://docs.influxdata.com/influxdb/v2/reference/api/>