



InfluxDB

Eine Einführung

Louis Beul | #588411 | HTW Berlin
B212: **Internet of Things** (SoSe 24)



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Agenda

1. Zeitreihen und Zeitreihendatenbanken

- a. Zeitreihen
- b. Zeitreihendatenbanken

2. InfluxDB in der Theorie

- a. Definition
- b. Datenmodell & Aufbau
- c. Points
- d. Line Protocol
- e. Flux Query Language

3. InfluxDB in der Praxis

- a. Erste Schritte
- b. Einbindung ins IoT-Projekt

4. Fragerunde

Einführung | Zeitreihendaten(banken)

www.htw-berlin.de



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

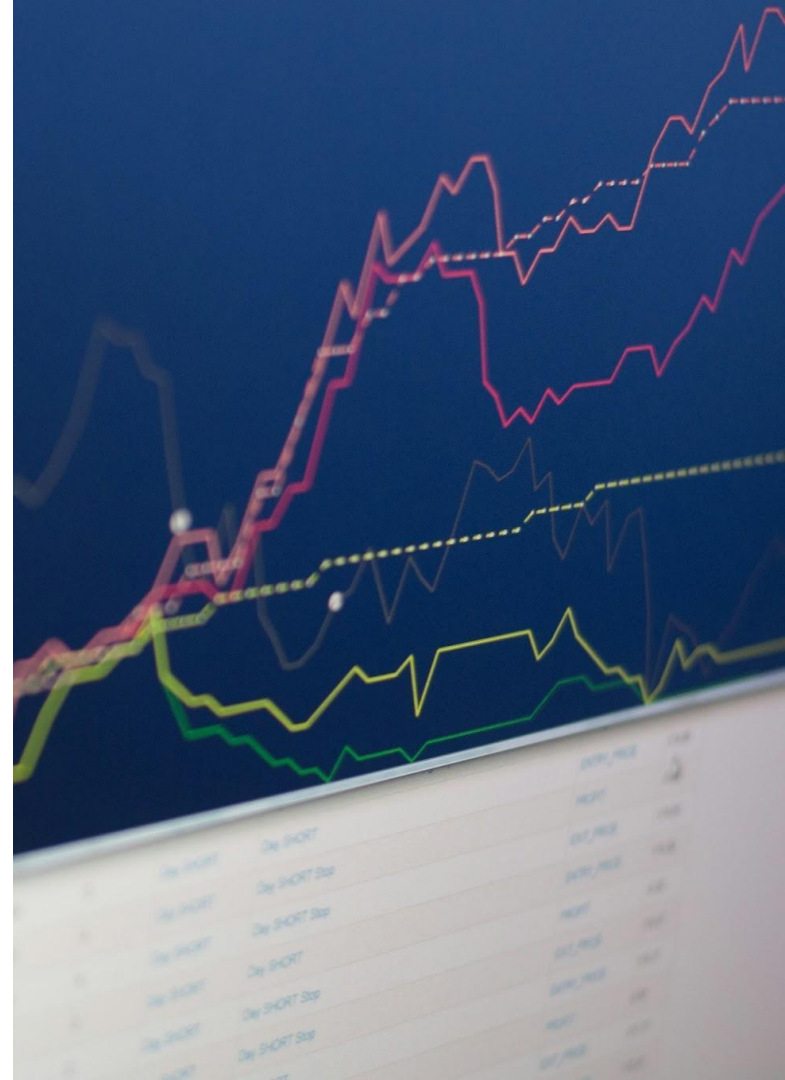
Zeitreihen | Was ist das?

- englisch **Time Series Data**
- **Chronologisch** geordnete Folge von Messungen
- Meist in regelmäßigen **Intervallen** gemessen
- Über einen gewissen **Zeitraum** hinweg
- Bei der Darstellung als Graph ist eine Achse immer die **Zeit**

- Oft wird zwischen **Ereignissen** und **Metriken** unterschieden
 - **Metrik** (Metric): Messung in regelmäßigen Intervallen
 - **Ereignis** (Event): Messung in unregelmäßigen Intervallen

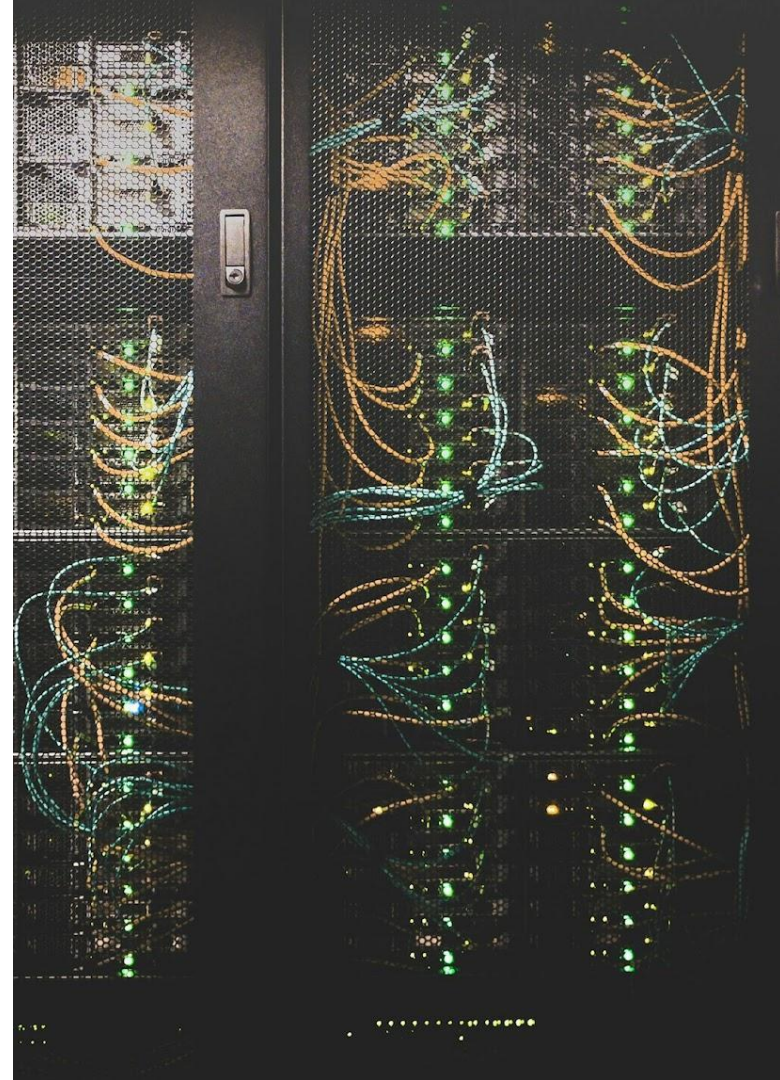
Zeitreihen | Beispiele

- Wertpapierkurse
- Herzfrequenz
- Wetterdaten
- Server Logs
- Network Traces
- Anruflhistorie
- **IoT-/Sensordaten**



Zeitreihen | Verarbeitung

- Verarbeitung in **Echtzeit**
- sehr **große Datenmengen**
- **Zeitstempel** statt klassischer IDs
- Fokus liegt auf **Schreiboperationen**
- fortlaufende **Aggregation**
- Anwendung:
 - Prognosen bzw. Trendanalysen
 - Monitoring
 - Machine Learning

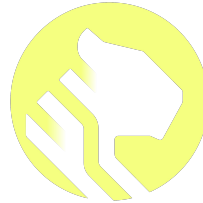


Zeitreihendatenbanken | Anforderungen

- Optimierte für häufige **Schreiboperationen**
- **Timestamp**-Indizierung
- Mechanismen zur **Datenlöschung** bzw. **-erhaltung**
- **Ressourcensparende** Speichermechanismen
 - Datenkompression
 - Downsampling
- Automatische, **kontinuierliche Queries**
- Einfache **Anbindungen** an Datenquellen
- Hohe **Skalierbarkeit**

Zeitreihendatenbanken | Beispiele

- **InfluxDB**
- Prometheus
- TimescaleDB
- DolphinDB
- Apache Druid



InfluxDB | Theorie

www.htw-berlin.de



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

InfluxDB | Definition

**“Time series analytics database
engineered for developers working
with real-time streaming data at
any scale”**

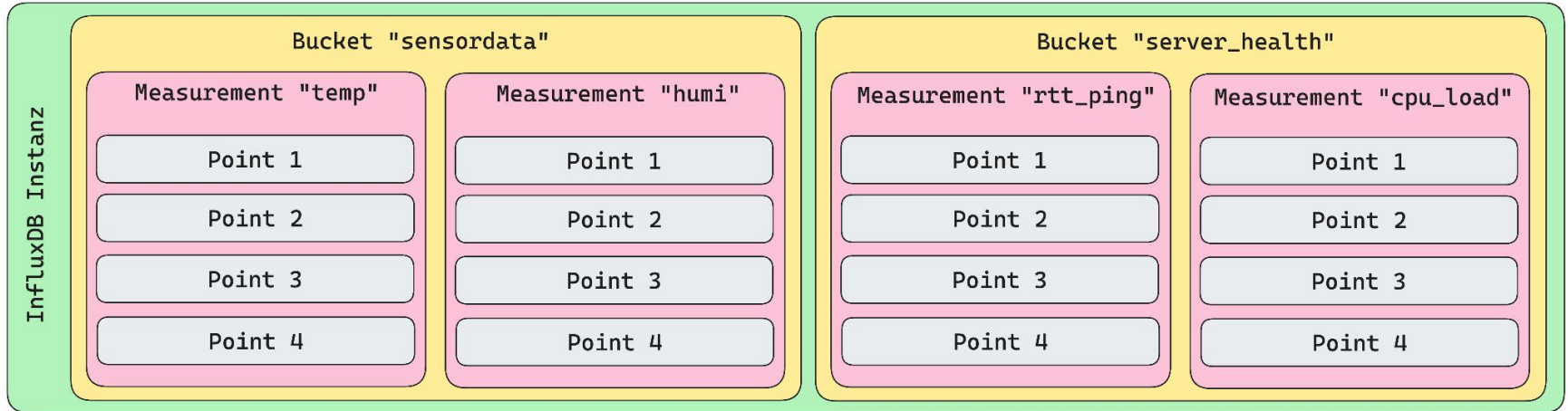
influxdata.com



InfluxDB | Datenmodell im Vergleich

Relationale DBMS (SQL)	InfluxDB (TSDB)
Datenbanken	Buckets
Tabellen	Measurements
Einträge bzw. Zeilen	Points
SQL als Query Language	Flux (Query Language)
Serialisierung via ORM (extern)	Serialisierung via Line Protocol
Keine automatische Löschung/Archivierung	Retention Policy

InfluxDB | Aufbau



InfluxDB | Points (Datensätze)

- Influx speichert einzelne Datensätze als **Points**
- Points haben immer 4 Bestandteile:
 - Das **Measurement** dem der Point angehört
(z.B. *humi*)
 - Ein **Tag Set** als Menge von indizierten Metadaten
(z.B. *studentId="911", location="C625"*)
 - Ein **Field Set** als Menge der eigentlichen Daten
(z.B. *value=39.66*)
 - Einen **TimeStamp** zur zeitlichen Einordnung



optional

InfluxDB | Line Protocol

- Das **Line Protocol** wird zum Serialisieren von Points genutzt
- Alle von Influx bereitgestellten **APIs, SDKs** und **Interfaces** benutzen intern dieses Protokoll
- Es ist sensitiv gegenüber **Leerzeichen**
- Tags sind immer als String formatiert
- Tag Sets und Field Sets können mehrere Einträge enthalten

measurement tag set field set timestamp

humi, room="625", student="411" value=39.6 00000000000000000000

Flux | Query Language

- **Flux** ist eine funktionale Skriptsprache zur Datenverarbeitung
- Gebaut für **InfluxDB**, funktioniert aber auch mit SQL, CSV, ...
- Flux Queries sind immer ähnlich aufgebaut:
 - a. Quelle spezifizieren
 - b. Zeitraum eingrenzen
 - c. Daten filtern
 - d. Daten verarbeiten, umformen oder aggregieren
 - e. Resultat zurückgeben
- Zeiträume können absolut oder relativ definiert werden
- Klassische Aggregatoren wie *min()*, *max()*, *mean()*, ...

Flux | Analogie Wasseraufbereitung



```
from(bucket:"sensor_data")  
  |> range(start: -24h)  
  |> filter(fn: (r) => r._measurement == "humi")  
  |> group(columns["studentId"])  
  |> mean()  
  |> yield(name: "mean_humi")
```

- Quelle angeben
- Zeitlich beschränken
- Filtern und aufbereiten
- Aggregieren
- Ausgeben

InfluxDB | Anbindungen

- Jede InfluxDB-Instanz kommt mit eingebauter **REST API**
 - Kommunikation via HTTP(S)
 - Requests enthalten Line Protocol oder Flux Queries
- Darüber hinaus gibt es viele **SDKs**
 - Python, Node, Arduino, Java, R, Scala, Swift ...
- Influx **Telegraf** als **Server-Agent** für weitere Schnittstellen
 - AWS, NginX, Jenkins, Salesforce, Redis, Windows OS ...
- Anbindung in **Node RED**
 - via node-red-contrib-influxdb Node
 - oder via HTTP API Request in JavaScript

InfluxDB | Retention, Tasks, Alerts

- Jeder neue Bucket bekommt eine **Retention Policy**
 - “Wie lange werden die Daten jeweils gespeichert?”
 - Standardmäßig unbegrenzt
- **Tasks** sind wiederkehrende, festgelegte Operationen
 - z.B. *“führe alle 24h den Query XYZ aus”*
 - oder *“exportiere alle 10 Tage den Bucket als CSV-Datei”*
- Influx kann eigene **Alerts** versenden
 - Basierend auf Task-Logik
 - Kann an jeglich HTTP Endpunkt angebunden werden
 - Vorkonfiguriert für Slack oder Pagerduty

InfluxDB | Praxis

www.htw-berlin.de



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

InfluxDB | Erste Schritte

- Influx installieren und starten
- Bucket mit Testdaten füllen
- Rumspielen und ausprobieren
 - Data Explorer
 - Dashboards
- Anleitung auf **GitHub**
 - Teil 01 selbst ausprobieren
 - Oder weiter zuschauen



github.com/LBeul/influx_workshop

InfluxDB | Integration ins Projekt

www.htw-berlin.de



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

InfluxDB | Einbindung ins Projekt

- Influx weiterhin laufen lassen
- Node RED starten
- Anleitung auf **GitHub** folgen
- Aufgeteilt in Read & Write Flows
 - Mit **Read** starten
 - Mit **Write** weiter machen
- Jetzt seid ihr an der Reihe
 - Queries auf eure Daten anpassen
 - Mit Flux herumspielen, etc.



github.com/LBeul/influx_workshop



**Vielen Dank.
Zeit für Eure Fragen!**

www.htw-berlin.de



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

Quellen

- <https://www.ibm.com/docs/de/spss-modeler/saas?topic=data-characteristics-time-series>
- <https://www.influxdata.com/what-is-time-series-data/>
- <https://www.influxdata.com/blog/influxdb-internals-101-part-one/>
- <https://docs.influxdata.com/influxdb/v1/concepts/glossary>
- <https://docs.influxdata.com/influxdb/v2/reference/syntax/line-protocol/>
- <https://flows.nodered.org/node/node-red-contrib-influxdb>
- <https://docs.influxdata.com/flux/v0/get-started/>
- https://github.com/LBeul/influx_workshop

Bildquellen

- <https://unsplash.com/> (lizenzfrei)
- <https://excalidraw.com/> (zur Diagrammerstellung)
- <https://docs.influxdata.com/flux/v0/get-started/> (Water Diagram)
- <https://www.the-qrcode-generator.com/>
- https://www.clipartmax.com/png/full/200-2006484_data-warehouse-apache-druid-logo.png
- <https://static-00.iconduck.com/assets.00/influxdb-icon-2017x2048-38lz5101.png>
- <https://www.timescale.com/static/1d1faec969cde90911998d145e43aaed/Time-scale-Brandmark-Yellow-PNG.png>
- https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Prometheus_software_logo.svg/2066px-Prometheus_software_logo.svg.png
- <https://dolphindb.gallerycdn.vsassets.io/extensions/dolphindb/dolphindb-vscode/2.0.1115/1706923515493/Microsoft.VisualStudio.Services.Icons.Default>
- https://www.influxdata.com/images/influxdata_full_navy-a7ca2ff4.svg



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de