



Studienarbeit - Entwurf

Ausarbeitung InfluxDB und Wetterdaten

Erstellt von:

Henrik Mertens
Hatzfelder Str 25
33104 Paderborn

Prüfer:

Prof. Dr. Ulrich. Reus

Eingereicht am:

31. Mai 2022

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Listingverzeichnis	VII
1 Einleitung	1
1.1 Zielsetzung	1
1.2 Aufbau und Vorgehensweise	1
2 Grundlagen	2
2.1 Time Series Data	2
2.2 Unterschiede zwischen Time Series und relationalen Datenbanken . . .	3
2.3 Verbreitete DBMS	3
2.4 Entwicklungsumgebung	5
2.4.1 Docker	5
2.4.2 Python und Jupyter Notebooks	5
2.4.3 InfluxDB Installation	5
3 InfluxDB	6
3.1 Daten einfügen	6
3.2 Daten abrufen	8
3.3 Daten verarbeiten und visualisieren	8
3.4 weitere InfluxDB-Funktionen	8
3.5 Wetterdaten	8
3.5.1 Wetterdaten Aufbau	8
3.5.2 Wetterdaten abrufen	9
4 Zusammenfassung	11
Anhang	12
Quellenverzeichnis	18

Ehrenwörtliche Erklärung

20

Abkürzungsverzeichnis

API	Application Programming Interface.
CLI	Command Line Interface.
CRUD	Create, Read, Update, Delete.
CSV	Comma-separated values.
DWD	Deutscher Wetterdienst.
HTML	Hypertext Markup Language.
HTTP	HyperText Transfer Protocol.
IOT	Internet of Things.
RDBMS	Relational Database Management System.
TSDB	Time Series Database.
UI	User Interface.

Abbildungsverzeichnis

Abbildung 1: DB-Engines Ranking	4
Abbildung 2: InfluxDB Dashboard	13
Abbildung 3: InfluxDB Load Data Source	14
Abbildung 4: InfluxDB Load Data Bucket	15
Abbildung 5: InfluxDB Load Data Bucket hinzufügen	15
Abbildung 6: InfluxDB Load Data API Tokens	16
Abbildung 7: InfluxDB Load Data API Token hinzufügen	17

Tabellenverzeichnis

Tabelle 1: Bedeutung der CSV Felder	10
Tabelle 2: Bedeutung Stationstabelle	10

Listingverzeichnis

Listing 1: InfluxDB Line Protokoll Quelle: Influxdata (2022b)	7
Listing 2: Wetterdaten CSV Quelle: DWD Wetterdaten CSV Download . .	9

1 Einleitung

1.1 Zielsetzung

Das Ziel dieser Arbeit ist es eine Einführung in die Funktion von Time Series Database (TSDB) zu geben. Außerdem soll beispielhaft an InfluxDB gezeigt werden, wie mit einer TSDB gearbeitet wird. Dazu werden die Wetterdaten des Deutschen Wetterdienst (DWD) importiert und ausgewertet.

1.2 Aufbau und Vorgehensweise

Im ersten Teil dieser Arbeit werden die Grundlagen von TSDB erklärt und Besonderheiten beschreiben. Im darauf folgenden Kapitel wird dann exemplarisch an InfluxDB gezeigt, wie mit einer TSDB gearbeitet wird. Im letzten Kapitel werden die Inhalte dieser Arbeit zusammengefasst.

2 Grundlagen

TSDB gehören zu den NoSQL Datenbanken und sind besonders darauf optimiert, mit Time Series Data zu arbeiten. Daruch können die große Mengen an Time Series Data verarbeiten durchsuchen und Speichern.¹

2.1 Time Series Data

Um TSDB zu verstehen, muss als erstes geklärt werden, was Time Series Data überhaupt ist und wie sie sich von anderen Daten unterscheiden. Wie der Name schon impliziert, ist Time Series Data eine Reihe von Daten die über einen Zeitraum gesammelt worden sind. Es wird also nicht nur der Endwert aufgezeichnet, sonder die Veränderung über einen Zeitraum. Diese Daten können z.B. Servermetriken, Netzwerkdaten, Internet of Things (IOT) Sensordaten, Ereignisse, Klicks, Marktgeschäfte und viele andere Arten von Daten sein. Time Series Data können gut daran erkannt werden, dass die Zeit eine wichtige Achse bei der Darstellung der Werte ist.²

Manchmal ist es nicht notwendig, alle Daten zu erfassen. Zum Beispiel wird in vielen Anwendungen nur der letzte Login gespeichert. Mehr ist auch für die Funktion nicht notwendig. Allerdings können zusätzliche Informationen gewonnen werden, wenn nicht nur die letzten Daten sondern die Veränderung aufgezeichnet wird. So kann zum Beispiel festgestellt werden, wie oft und wann sich der Kunde einloggt und ob es dabei ein Muster gibt. Anhand dieser Daten können Kunden dann kategorisiert werden.³

Eine Besonderheit von Time Series Data ist, dass sie sich nicht verändert. Wenn die Daten einmal erfasst wurden wird an ihnen nichts mehr verändert. Es werden nur neue Daten hinzugefügt.⁴

¹vgl. ComputerWeekly.de, Redaktion (2021)

²vgl. Dix, Paul (2021), S. 1 ff.

³vgl. Data-Science-Team (2020)

⁴vgl. Fangman, Sam (2019)

2.2 Unterschiede zwischen Time Series und relationalen Datenbanken

Um Time Series Data zu speichern, ist es nicht unbedingt erforderlich, eine TSDB zu nutzen. Auch relationale Datenbanken können Time Series Data speichern. Einer der wichtigsten Unterschiede zwischen einer TSDB im Gegensatz zu einem Relational Database Management System (RDBMS) ist es, dass kein Datenbank Schema notwendig ist. Wenn Time Series Daten in einer Relationalen Datenbank geschrieben werden sollen, müssen erst entsprechende Tabellen angelegt werden, in denen die Daten immer im gleichen Format abgelegt werden müssen. Im Gegensatz dazu können in einer TSDB die Daten einfach schemafrei in die Datenbank geschrieben werden. Ein weiterer Vorteil ist, dass TSDB im Gegensatz zu relationalen Datenbanken besser und einfacher skaliert werden können.⁵

Aber TSDB haben nicht nur Vorteile. Wie in Abb. 1 zu sehen, sind sie viel weniger verbreitet als nicht zeit basierte Datenbank Systeme. Dadurch gibt es viel weniger Entwickler die sich mit TSDB auskennen und auch das Ökosystem um die Datenbank ist deutlich kleiner. Außerdem sind RDBMS dadurch, dass es sie viel länger gibt, sehr stabil und sehr gut unterstützt.⁶

RDBMS arbeiten nach dem Create, Read, Update, Delete (CRUD) Prinzip, welches für Time Series Data nicht optimal ist. Auf Time Series Data werden keine Update Befehle durchgeführt, da neue Daten immer nur als neuer Datenpunkt angehängt werden. Auch das Löschen von Daten wird nicht sehr häufig durchgeführt und im Gegensatz zu RDBMS meistens gleichzeitig auf einer großen Menge an Datensätzen. Daher sind TSDB besser dafür geeignet, mit Time Series Data zu arbeiten und weisen auch eine höhere Performance auf.⁷

2.3 Verbreitete DBMS

Aktuell gibt es wie in Abb. 1 zu sehen, einige beliebte Multi-Model Datenbanken, die als TSDB genutzt werden können. So können die Datenbanken MongoDB, Redis, Teradata

⁵vgl. Influxdata (2021)

⁶vgl. Influxdata (2021)

⁷vgl. Influxdata (2021)

und Couchbase mit Time Series Daten arbeiten. Die erste reine TSDB im Ranking ist InfluxDB auf Platz 29.⁸

Abbildung 1: DB-Engines Ranking

394 Systeme im Ranking, Mai 2022

Rang			DBMS	Datenbankmodell	Punkte		
Mai 2022	Apr 2022	Mai 2021			Mai 2022	Apr 2022	Mai 2021
1.	1.	1.	Oracle +	Relational, Multi-Model	1262,82	+8,00	-7,12
2.	2.	2.	MySQL +	Relational, Multi-Model	1202,10	-2,06	-34,28
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-Model	941,20	+2,74	-51,46
4.	4.	4.	PostgreSQL +	Relational, Multi-Model	615,29	+0,83	+56,04
5.	5.	5.	MongoDB +	Document, Multi-Model	478,24	-5,14	-2,78
6.	6.	7.	Redis +	Key-value, Multi-Model	179,02	+1,41	+16,85
7.	8.	6.	IBM Db2	Relational, Multi-Model	160,32	-0,13	-6,34
8.	7.	8.	Elasticsearch +	Suchmaschine, Multi-Model	157,69	-3,14	+2,34
9.	9.	10.	Microsoft Access	Relational	143,44	+0,66	+28,04
10.	10.	9.	SQLite +	Relational	134,73	+1,94	+8,04
11.	11.	11.	Cassandra +	Wide column	118,01	-3,98	+7,08
12.	12.	12.	MariaDB +	Relational, Multi-Model	111,13	+0,81	+14,44
13.	13.	13.	Splunk	Suchmaschine	96,35	+1,11	+4,24
14.	14.	27.	Snowflake +	Relational	93,51	+4,06	+63,46
15.	15.	15.	Microsoft Azure SQL Database	Relational, Multi-Model	85,33	-0,45	+14,88
16.	16.	16.	Amazon DynamoDB +	Multi-Model	84,46	+1,55	+14,39
17.	17.	14.	Hive +	Relational	81,61	+0,18	+5,42
18.	18.	17.	Teradata +	Relational, Multi-Model	68,39	+0,82	-1,59
19.	19.	19.	Neo4j +	Graph	60,14	+0,62	+7,91
20.	20.	20.	Solr	Suchmaschine, Multi-Model	57,26	-0,48	+6,07
21.	21.	18.	SAP HANA +	Relational, Multi-Model	55,09	-0,71	+2,33
22.	22.	22.	FileMaker	Relational	52,27	-0,64	+5,55
23.	24.	24.	Google BigQuery +	Relational	48,61	+0,63	+10,98
24.			Databricks	Multi-Model	47,85		
25.	23.	21.	SAP Adaptive Server	Relational, Multi-Model	47,78	-0,58	-2,19
26.	25.	23.	HBase +	Wide column	43,19	-1,14	-0,05
27.	26.	25.	Microsoft Azure Cosmos DB +	Multi-Model	40,22	-0,12	+5,51
28.	27.	28.	PostGIS	Spatial DBMS, Multi-Model	31,82	-0,23	+1,98
29.	28.	29.	InfluxDB +	Time Series, Multi-Model	29,55	-0,47	+2,38
30.	29.	26.	Couchbase +	Document, Multi-Model	28,38	-0,67	-1,85

Quelle: <https://db-engines.com/de/ranking?msclkid=4f2a29e5d08811ec95ccd74f8f5146ab>

Allerdings haben Datenbanken, die nur auf das Verarbeiten von Time Series Data ausgelegt sind, deutliche Performance Vorteile gegenüber Multi Model Datenbanken. In einem Vergleich von InfluxDB und MongoDB, hat InfluxDB eine 2,4 mal bessere Schreibperformance als MongoDB und ist beim Lesen sogar 5,7 mal schneller. InfluxDB benötigt außerdem 20 mal weniger Speicherplatz auf der Festplatte, um die gleiche Menge an Daten zu speichern.⁹

⁸vgl. solid-IT-gmbh (2022)

⁹vgl. Hajek Vlasta Pour Ales, Kudibal Ivan (2019)

2.4 Entwicklungsumgebung

2.4.1 Docker

2.4.2 Python und Jupyter Notebooks

2.4.3 InfluxDB Installation

3 InfluxDB

InfluxDB ist eine in Go geschriebene open source TSDB, die darauf ausgelegt ist, mit einer großen Menge an Time Series Data zu arbeiten.¹⁰ Im weiteren Verlauf dieses Kapitels wird am Beispiel von Wetterdaten gezeigt, wie mit InfluxDB gearbeitet wird. InfluxDB stellt für die Integration in eigene Anwendungen ein HyperText Transfer Protocol (HTTP) Application Programming Interface (API) zur Verfügung, für die es in vielen Programmiersprachen Client Librarys gibt. Außerdem wird ein Webinterface und ein Command Line Interface (CLI) bereitgestellt.¹¹

3.1 Daten einfügen

In InfluxDB werden Daten immer in Buckets gespeichert. Um Daten hochzuladen, muss zunächst ein Bucket angelegt werden. Dazu gibt es zwei Möglichkeiten. Die Einfachste ist es, über das Web User Interface (UI) von InfluxDB einen neuen Bucket anzulegen. Dazu muss nach dem Login der Navigationspunkt Data und der Reiter Buckets ausgewählt werden. Hier kann dann mit dem Button Create Bucket ein neuer Bucket angelegt werden. Beim Anlegen kann noch eine Lebensdauer für die Daten ausgewählt werden, nach welcher die jeweiligen Datenpunkte gelöscht werden.¹²

Daten werden immer nach dem InfluxDB Line Protokoll formatiert an die Datenbank gesendet. Das Protokoll ist wie in Listing 1 dargestellt aufgebaut. Im ersten Teil des Line Protokolls wird der Name der Messreihe angegeben. Das kann zum Beispiel der Name des Sensors sein oder der Ort an dem der Messwert genommen wurde. Wichtig ist, dass Groß und Kleinschreibung beachtet werden muss und Unterstriche nicht genutzt werden dürfen. Sonderzeichen müssen mit einem \ maskiert werden. Nach dem Namen kommen getrennt durch ein Komma die Tags der Messung. Tags werden indexiert und dazu genutzt, um Messwerte zu durchsuchen. Tags werden als Key Value Paar angegeben. Hier sollen Metadaten wie zum Beispiel der Standort des Sensors oder der Name des Servers eingetragen werden, zu dem die Datenpunkt/e gehören. Die eigentlichen Werte sind mit einem Leerzeichen von den Tags abgegrenzt und bestehen aus durch Kommas getrennten Key Value Feldern. Der letzte Wert einer Zeile ist der

¹⁰vgl. solid-IT-gmbh (2022)

¹¹vgl. Influxdata (2022a)

¹²vgl. Abb. 2, Abb. 3, Abb. 4, Abb. 5

Unix Timestamp in Millisekunden. In einer Datei oder Anfrage kann es mehrere Zeilen mit Daten geben.¹³

Listing 1: InfluxDB Line Protokoll Quelle: Influxdata (2022b)

measurementName	tagKey=tagValue	fieldKey="fieldValue"	1465839830100400200
Measurement	Tag set	Field set	Timestamp

Die im Line Protokoll formatierten Daten können jetzt entweder mithilfe eines Rest Requests oder des InfluxDB CLI in die Datenbank übertragen werden. Um diese Anfragen zu autorisieren muss ein API Token mitgesendet werden.¹⁴ Um einen Token zu bekommen, kann dieser entweder über das Webinterface, die CLI oder über die API angelegt werden. Der einfachste Weg ist es, den Token über das Webinterface anzulegen. Dazu wird wie beim anlegen eines Buckets zunächst der Menüpunkt Data ausgewählt und anschließend der Reiter API Tokens. Mit einem Klick auf Generate API Token kann dann ein API Token erstellt werden.¹⁵ Dabei kann zwischen einem All-Access token und einem Read/Write token ausgewählt werden. Mit dem All Access Token kann auf alles zugegriffen werden. Mit einem Read/Write Token kann wie in Abb. 7 zu sehen, ausgewählt werden, auf welchen Bucket geschrieben oder gelesen werden kann.

16

¹³vgl. Influxdata (2022b)

¹⁴vgl. Influxdata (2022d)

¹⁵vgl. Abb. 2, Abb. 3, Abb. 6, Abb. 7

¹⁶vgl. Influxdata (2022c)

3.2 Daten abrufen

3.3 Daten verarbeiten und visualisieren

3.4 weitere InfluxDB-Funktionen

3.5 Wetterdaten

3.5.1 Wetterdaten Aufbau

Die Wetterdaten des DWD können über den CDC OpenData Bereich heruntergeladen werden. Hier werden die Wetterdaten über FTP und HTTPS zum Download angeboten. Unter der URL https://www.dwd.de/DE/leistungen/cdc/cdc_ueberblick-klimadaten.html wird eine gute Übersicht über die zum Download angebotenen Daten geboten. Die Werte für die aktuelle Lufttemperatur können über https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/10_minutes/air_temperature/now/ abgerufen werden. Historische Daten können über https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/10_minutes/air_temperature/now/ abgerufen werden.

Aktuell werden auf der Webseite, für die aktuelle Lufttemperatur, ca 480 Dateien zum Download angeboten. Die meisten dieser Dateien entsprechen jeweils einer Messstation und je nach Tageszeit kann deswegen die Menge der Zeilen in der Datei variieren, weil immer um 00:00 eine neue Datei angefangen wird. In den Zip-Dateien finden sich außerdem Metadaten über die Messstationen. Die eigentlichen Daten sind als Comma-separated values (CSV) formatiert und sehen aus wie in Listing 2 gekürzt dargestellt. In der CSV Datei gibt es 9 Felder. Der Inhalt der Felder wird in Tabelle 1 beschrieben.

In der Datei `zehn_now_tu_Beschreibung_Stationen.txt` werden die Wetterstationen beschrieben. Diese Datei ist nicht als CSV Datei formatiert sondern als Tabelle und erhält Daten über die Wetterstationen. Die Daten der Stationen in der heruntergeladenen Textdatei stimmen mit den Daten der Hauptamtliches Messnetz Karte überein. Allerdings enthält die Textdatei nicht alle Stationen sondern nur Station für die auch Messwerte im Datensatz hinterlegt sind. Die Bedeutung der einzelnen Spalten der Tabelle sind in der Tabelle 2 beschreiben.

Listing 2: Wetterdaten CSV Quelle: DWD Wetterdaten CSV Download

```

STATIONS_ID;MESS_DATUM; QN;PP_10;TT_10;TM5_10;RF_10;TD_10;eor
73;202205120000; 2; -999; 12.9; 11.2; 84.2; 10.3;eor
73;202205120010; 2; -999; 12.7; 11.2; 84.9; 10.2;eor
73;202205120020; 2; -999; 12.9; 11.4; 83.0; 10.1;eor
73;202205120030; 2; -999; 12.4; 10.7; 86.9; 10.3;eor
73;202205120040; 2; -999; 12.4; 10.5; 86.2; 10.2;eor
73;202205120050; 2; -999; 12.3; 10.3; 85.5; 9.9;eor
73;202205120100; 2; -999; 12.1; 10.1; 88.1; 10.2;eor
73;202205120110; 2; -999; 11.7; 9.9; 90.1; 10.1;eor
73;202205120120; 2; -999; 11.7; 10.0; 89.0; 10.0;eor

```

3.5.2 Wetterdaten abrufen

Um die Daten auswerten zu können müssen diese als erstes heruntergeladen und entpackt werden. Dazu wird mithilfe von BeautifulSoup aus der Hypertext Markup Language (HTML) Seite des DWD für jede Datei eine URL ausgelesen. Die so gewonnen URLs können dann mithilfe einer Schleife heruntergeladen werden. Um den Messwerten eine Station zuzuordnen zu können wird als erstes die Datei mit den Station verarbeitet. Für jede Station wird Objekt erstellt und in ein dictionary gespeichert. Dadurch kann in im dictionary einfach über die STATIONS_ID die passende Station gefunden werden. Weil diese Datei allerdings nicht CSV formatiert ist muss die Datei auf eine andere Art ausgewertete werden. Um die einzelnen Felder aus einer Zeile zu bekommen wird also immer so lange gelesen bis wieder ein bestimmte Anzahl von Leerzeichen hintereinander erkannt worden ist. Die Zeichen zwischen den Leerzeichen sind dann ein ausgelesenes Feld. Nachdem die Stationsdatei ausgewertet word ist werden die CSV Dateien in einer Schleife entpackt und mithilfe der Bibliothek Pandas in ein Dataframe umgewandelt. Das so erzeugte Dataframe wir im letzten Schritt mit den Daten der Stations zusammengeführt und als Datenpunkt in InfluxDB geschrieben.

Weitere Erklärungen und der Code selbst kann im angehängten Jupyter notebook eingesehen und ausgeführt werden.

Tabelle 1: Bedeutung der CSV Felder

Feld Name	Bedeutung
STATION_ID	Gibt an von welcher Station die Werte stammen
MESS_DATUM	Gibt an wann gemessen wurde im Format %Y%m%d%H%M. Also Jahr Monat Tag Stunde Minute als eine zusammengeschrriebene Zahl.
QN	Gibt die Qualitt der Messwerte an. Hier gibt es die Werte 1 bis 3 1. nur formale Kontrolle bei Dekodierung und Import 2. Kontrolle mit individuell festgelegten Kriterien 3. ROUTINE automatische Kontrolle und Korrektur mit Software (QUALIMET)
PP_10	Luftdruck auf Stationshhe
TT_10	Lufttemperatur auf 2m Hhe
TM5_10	Lufttemperatur auf 5cm Hhe
TD_10	relative Luftfeuchtigkeit auf 2m Hhe
eor	END OF RECORD"Bededeutet die Zeile ist zu Ende.

Quelle: Eigene Darstellung https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/10_minutes/air_temperature/now/BESCHREIBUNG_obsgermany_climate_10min_tu_now_de.pdf

Tabelle 2: Bedeutung Stationstabelle

Feld Name	Bedeutung
STATION_ID	Gibt an von welcher Station die Werte stammen
von_datum	Datum seit dem die Station aktiv ist.
bis_datum	Hohe der Station.
Stationshoehe	Hohe ber dem Normalnullpunkt
geoBreite	Breitengrad der Station.
geoLaenge_10	Lngengrad der Stations.
Stationsname	Name der Station.
Bundesland	Bundesland in dem die Station steht.

Quelle: Vergleich der Werte mit der Hauptamtliches Messnetz Karte https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/10_minutes/air_temperature/now/zehn_now_tu_Beschreibung_Stationen.txt

4 Zusammenfassung

Dieses Dokument ist eine Hilfe, um die Formalien für eine Bachelor-Thesis an der FHDW bei der Verwendung von $\text{\LaTeX}$ zu erfüllen und dabei möglichst viele Automatismen von $\text{\LaTeX}$ zu nutzen. Eine Absprache mit dem betreuenden Professor ist dennoch ratsam.

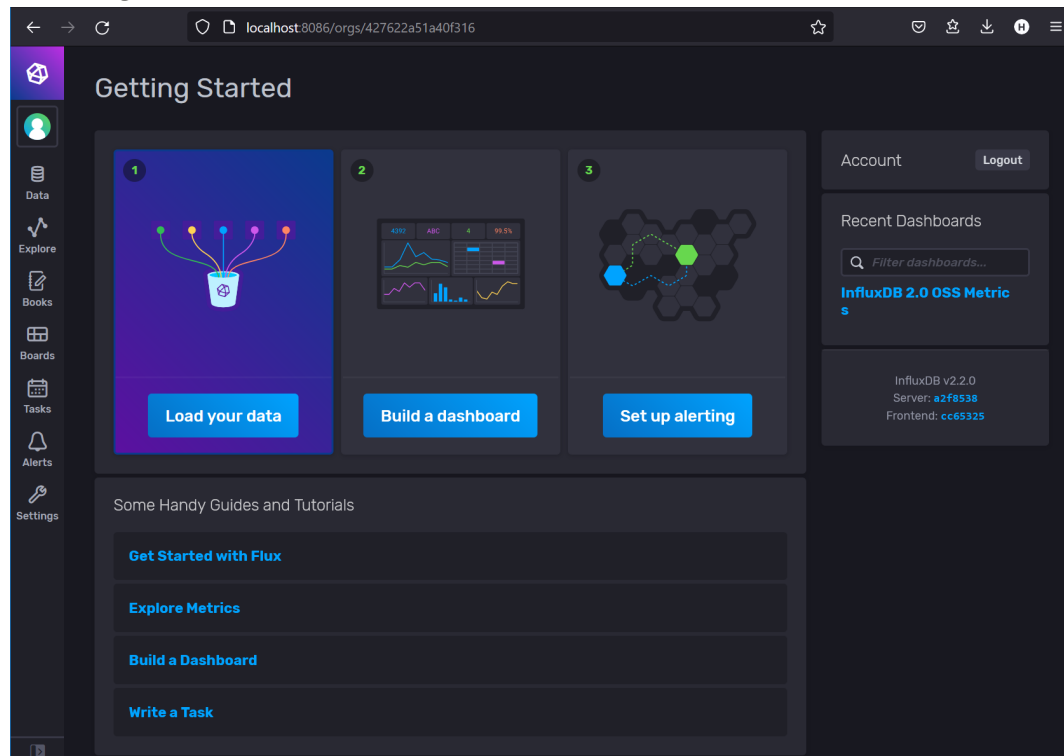
Anhang

Anhangsverzeichnis

Anhang 1: InfluxDB Webinterface Screenshot	13
--	----

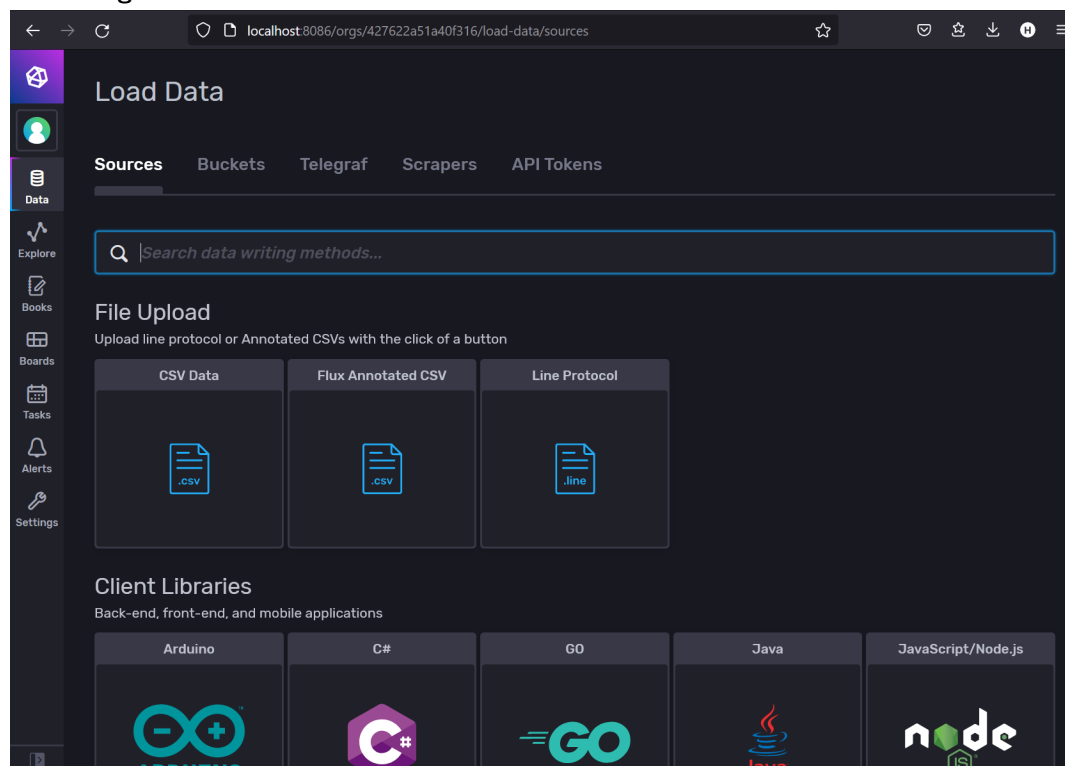
Anhang 1 InfluxDB Webinterface Screenshot

Abbildung 2: InfluxDB Dashboard



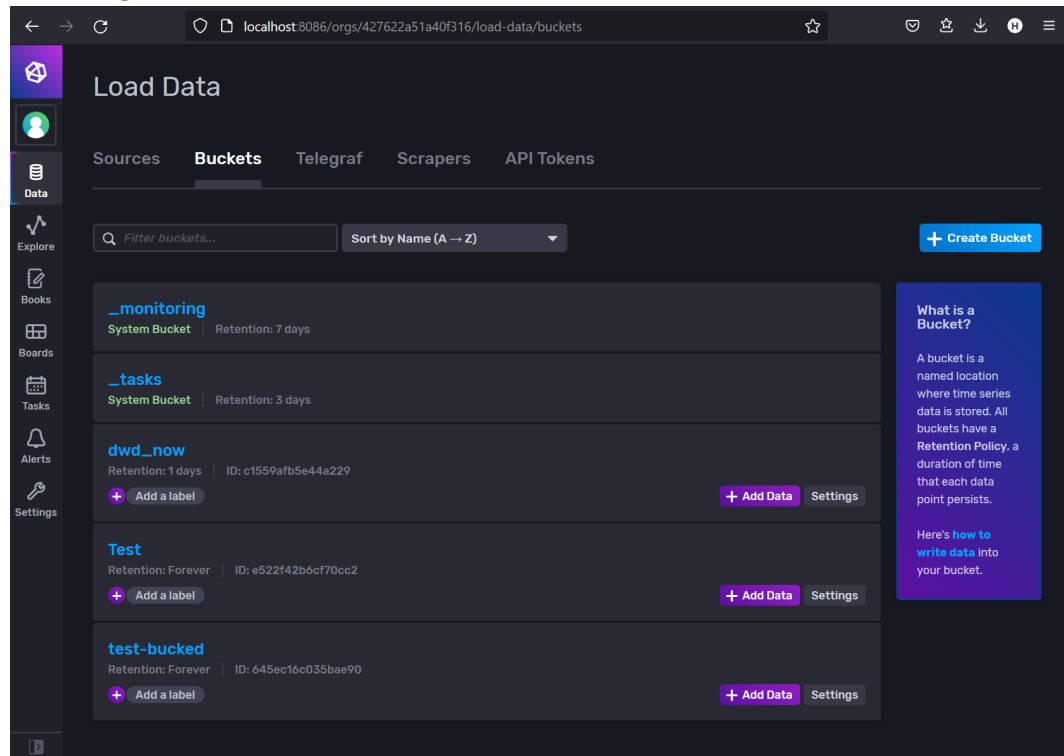
Quelle: Eigener Screenshot InfluxDB Webinterface

Abbildung 3: InfluxDB Load Data Source



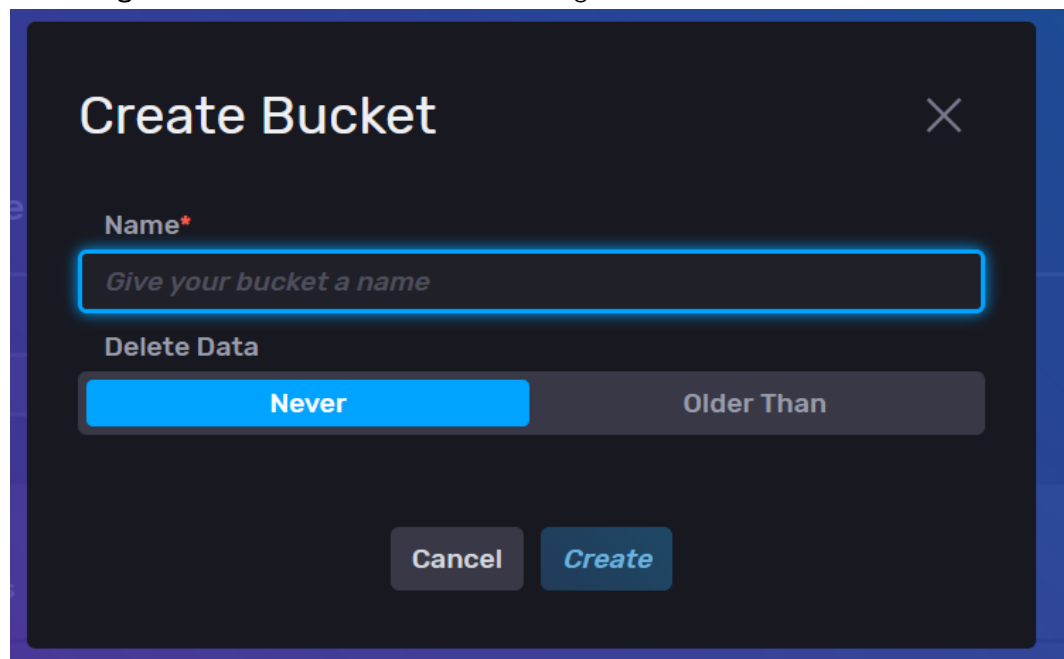
Quelle: Eigener Screenshot InfluxDB Webinterface

Abbildung 4: InfluxDB Load Data Bucket

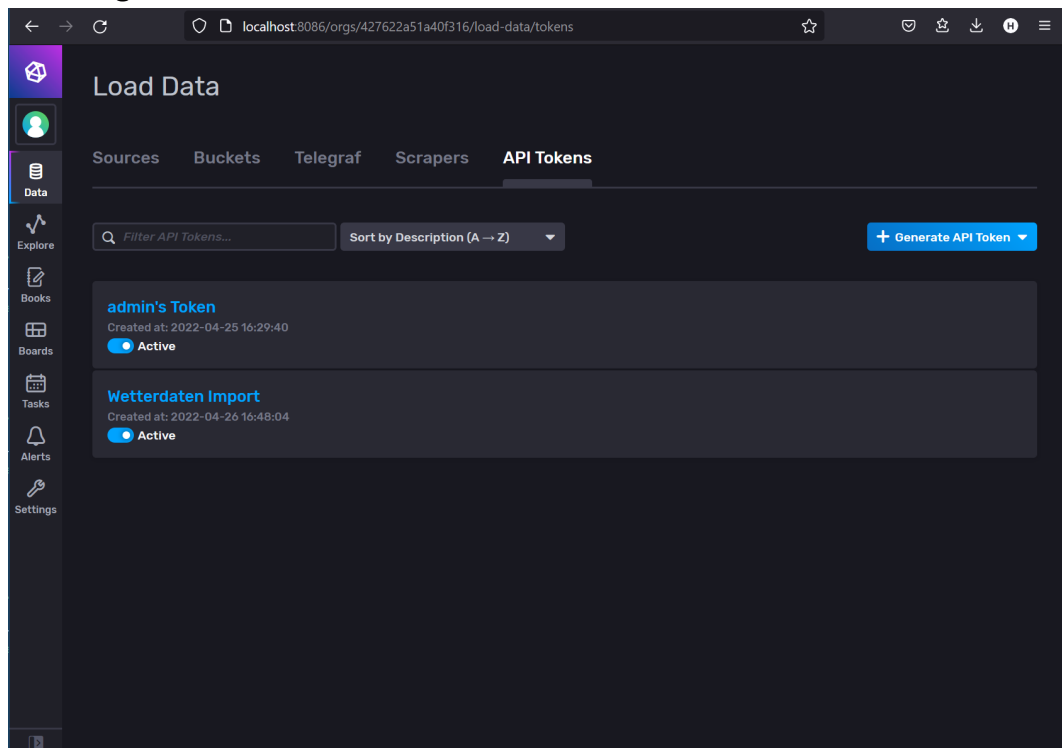


Quelle: Eigener Screenshot InfluxDB Webinterface

Abbildung 5: InfluxDB Load Data Bucket hinzufügen



Quelle: Eigener Screenshot InfluxDB Webinterface

Abbildung 6: InfluxDB Load Data API Tokens

Quelle: Eigener Screenshot InfluxDB Webinterface

Abbildung 7: InfluxDB Load Data API Token hinzufügen

Generate Read/Write API Token

Description

Describe this new API token

Read

All Buckets **Scoped**

BUCKETS Select All Deselect All

Search buckets...

Test
dwd_now
test-bucked

Write

All Buckets **Scoped**

BUCKETS Select All Deselect All

Search buckets...

Test
dwd_now
test-bucked

Cancel Save

Quelle: Eigener Screenshot InfluxDB Webinterface

Quellenverzeichnis

Internetquellen

ComputerWeekly.de, Redaktion (2021). *Definition Zeitreihendatenbank (Time Series Database, TSDB)*. URL: <https://datascience.eu/wiki/what-the-heck-is-time-series-data-and-why-do-i-need-a-time-series-database/> (besucht am 21. Mai 2021).

Data-Science-Team (2020). *What the heck is time-series data (and why do I need a time-series database)?* URL: <https://www.computerweekly.com/de/definition/Zeitreihendatenbank-Time-Series-Database-TSDB> (besucht am 9. Mai 2022).

Dix, Paul (2021). *Why Time Series Matters for Metrics, Real-Time Analytics and Sensor Data*. URL: <http://get.influxdata.com/rs/972-GDU-533/images/why%20time%20series.pdf> (besucht am 10. Mai 2022).

Fangman, Sam (2019). *The Time Has Come for a New Type of Database*. URL: <https://medium.datadriveninvestor.com/the-time-has-come-for-a-new-type-of-database-47cf8df1667a> (besucht am 10. Mai 2022).

Hajek Vlasta Pour Ales, Kudibal Ivan (2019). *Why Time Series Matters for Metrics, Real-Time Analytics and Sensor Data*. URL: <http://get.influxdata.com/rs/972-GDU-533/images/why%20time%20series.pdf> (besucht am 27. Mai 2022).

Influxdata (2021). *Why You Should Migrate from SQL to NoSQL for Time Series Data*. URL: <https://www.influxdata.com/from-sql-to-nosql/> (besucht am 20. Mai 2022).

Influxdata (2022a). *API Quick Start*. URL: https://docs.influxdata.com/influxdb/v2.2/api-guide/api_intro/ (besucht am 21. Mai 2022).

Influxdata (2022b). *Line protocol*. URL: <https://docs.influxdata.com/influxdb/v2.2/reference/syntax/line-protocol/> (besucht am 29. Mai 2022).

Influxdata (2022c). *Manage API tokens*. URL: <https://docs.influxdata.com/influxdb/cloud/security/tokens/#all-access-token> (besucht am 29. Mai 2022).

Influxdata (2022d). *Write data with the InfluxDB API*. URL: <https://docs.influxdata.com/influxdb/v2.2/write-data/developer-tools/api/> (besucht am 29. Mai 2022).

solid-IT-gmbh (2022). *DB-Engines Ranking*. URL: <https://db-engines.com/de/ranking> (besucht am 10. Mai 2022).

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Studienarbeit - Entwurf selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Paderborn, 31. Mai 2022

Henrik Mertens