hochschule hof
University of Applied Sciences

Non-Relational Databases

InfluxDB

Prof. Dr. Jürgen Heym

Hof University of Applied Sciences

1 – Introduction

hochschule hof
University of Applied Sciences

- InfluxDB is developed by InfluxData
  https://www.influxdata.com/

- Key features
  - ✓ open source
  - ✓ big data
  - ✓ NoSQL database
  - ✓ massive scalability
  - ✓ high availability
  - ✓ fast read & write
  - ✓ time-series database
  - ✓ supports regular time-series data (e.g. heart beat)
  - ✓ support irregular time-series data (e.g. discrete events)
  - ✓ SQL like query language

1 – Introduction

hochschule hof
University of Applied Sciences

- Supported API languages
  - ✓ Go
  - ✓ Java
  - ✓ Python
  - ✓ PHP
  - ✓ Node.js

- Use cases
  - ✓ DevOps real-time monitoring
  - ✓ Internet of Things (IoT) and Industrial IoT (IIoT) monitoring
  - ✓ Time-series based analytics applications
  - ✓ Geo positioning and tracking

2 – Key Concepts and Terms

- InfluxDB data organization
  - ✓ Database --- similar to SQL database
  - ✓ Measurement --- similar to SQL tables
  - ✓ Tags --- similar to SQL table indexed columns e.g. ticker, company, time
  - ✓ Fields --- similar to SQL table with unindexed columns e.g. close, high, low, open
  - ✓ Point of events --- similar to SQL rows

InfluxDB example: measurement tickers

```
name: tickers
tags: company=Alphabet Inc, ticker=GOOGL
time                 close   high    low     open    volume
----                 -----   ----    ---     ----    ------
2017-11-22T14:30:00Z 1051.16 1051.16 1051.16 1051.16 13622
2017-11-22T14:31:00Z 1051.46 1051.78 1051.11 1051.42 1404
2017-11-22T14:32:00Z 1051.07 1051.57 1050.72 1051.53 7014
2017-11-22T14:33:00Z 1051.35 1051.53 1050.61 1051    5973
2017-11-22T14:34:00Z 1051.33 1051.75 1051.27 1051.75 1500

name: tickers
tags: company=Apple Inc, ticker=AAPL
time                 close  high   low    open   volume
----                 -----  ----   ---    ----   ------
2017-11-22T14:30:00Z 173.45 173.45 173.34 173.36 300218
2017-11-22T14:31:00Z 173.7  173.71 173.35 173.39 165442
2017-11-22T14:32:00Z 173.56 173.8  173.51 173.7  175809
2017-11-22T14:33:00Z 173.62 173.75 173.56 173.57 109326
2017-11-22T14:34:00Z 173.91 173.93 173.61 173.62 218830
```

SQL example: table tickers

time	close	company	high	low	open	ticker	volume
2017-11-22T14:30:00Z	1051.16	Alphabet Inc	1051.16	1051.16	1051.16	GOOGL	13622
2017-11-22T14:30:00Z	173.45	Apple Inc	173.45	173.34	173.36	AAPL	300218
2017-11-22T14:31:00Z	1051.46	Alphabet Inc	1051.78	1051.11	1051.42	GOOGL	1404
2017-11-22T14:31:00Z	173.70	Apple Inc	173.71	173.35	173.39	AAPL	165442
2017-11-22T14:32:00Z	1051.07	Alphabet Inc	1051.57	1050.72	1051.53	GOOGL	7014
2017-11-22T14:32:00Z	173.56	Apple Inc	173.80	173.51	173.70	AAPL	175809
2017-11-22T14:33:00Z	1051.35	Alphabet Inc	1051.53	1050.61	1051.00	GOOGL	5973
2017-11-22T14:33:00Z	173.62	Apple Inc	173.75	173.56	173.57	AAPL	109326
2017-11-22T14:34:00Z	1051.33	Alphabet Inc	1051.75	1051.27	1051.75	GOOGL	1500
2017-11-22T14:34:00Z	173.91	Apple Inc	173.93	173.61	173.62	AAPL	218830

2 – Key Concepts and Terms

hochschule hof
University of Applied Sciences

- InfluxDB terms
  - ✓ Measurement === similar to SQL tables
  - ✓ Fields / Field set === collection of field keys and field values on a point, similar to unindexed SQL columns
  - ✓ Field key === string type, stores metadata
  - ✓ Field value === stores your data
  - ✓ Tags / Tag set === collection of tag keys and tag values on a point, optional, similar to indexed SQL columns
  - ✓ Tag key === string type, stores metadata
  - ✓ Tag value === string type, stores metadata
  - ✓ Continuous query === similar to SQL stored procedure
  - ✓ Line protocol === text-based line format for writing points to InfluxDB
  - ✓ Point === corresponds to a single row in a SQL database

2 – Key Concepts and Terms

hochschule hof
University of Applied Sciences

- InfluxDB terms continued
  - ✓ Retention policy (RP) === describes the duration of data points
    how many copies are stored in the cluster
    each RP is unique in the database
    default RP is “autogen”

```
name    duration shardGroupDuration replicaN default
----    -------- ------------------ -------- -------
autogen 0s       168h0m0s           1        true
```

  - ✓ Series === collection of data in InfluxDB
    share the same RP, measurement and tag set
  - ✓ Timestamps === stored in RFC 3339 UTC format
  - ✓ Time structured Merge tree (TSM) === InfluxDB storage engine
  - ✓ Write Ahead Log (WAL) === temporary cache, allows for efficient batching of writes

3 – Data Model and Storage Engine

hochschule hof
University of Applied Sciences

- Which type of data is stored in InfluxDB ?
  - ✓ Sequences of data points based on time
  - ✓ Hundred s of millions data points
  - ✓ Data points are typically immutable & read only
  - ✓ New points are appended to a measurement
- Which design for the data mode ?
  - ✓ Which attributes should we use for indexed tags ?
  - ✓ Which attributes should we use for unindexed fields ?

3 – Data Model and Storage Engine

hochschule hof
University of Applied Sciences

- Design recommendations

  - ✓ Frequently searched data === tag
  - ✓ Needs a **`group by`** statement === tag
  - ✓ Data used in InfluxDB functions === field
  - ✓ Data are of non-string type === field
  - ✓ Data have dynamic values === field

  - ✓ Avoid too many series (same RP, measurement & tag set)
  - ✓ Avoid tags contain large string information => high memory usage
  - ✓ Avoid highly dynamic tag values => high memory usage

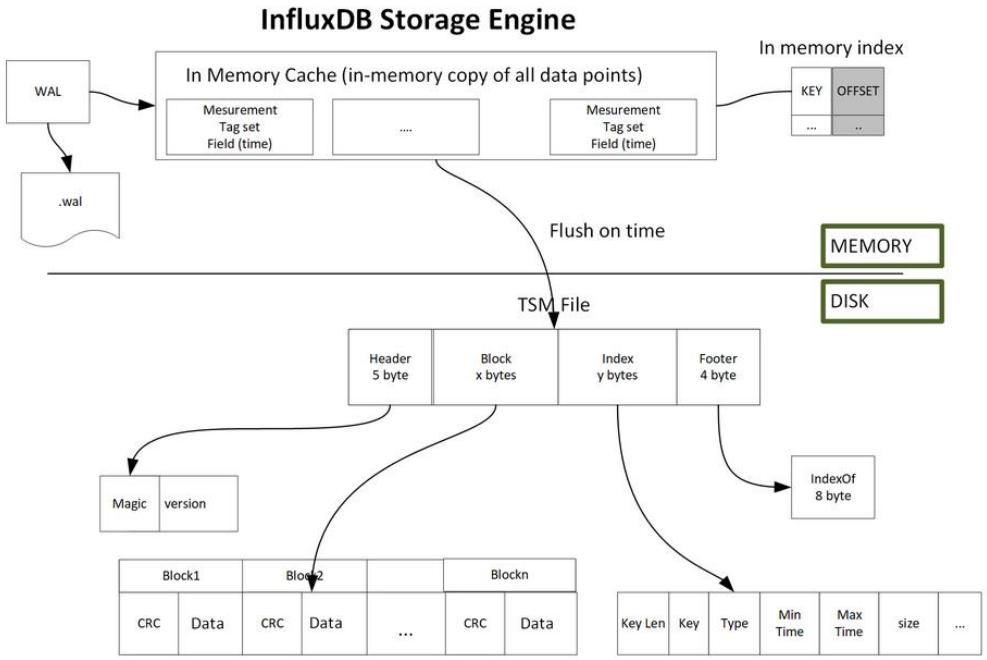
  - ✓ Consult InfluxDB hardware guidelines for further InfluxDB design recommendations !

4 – Storage Engine

hochschule hof
University of Applied Sciences

- TSM tree

  - ✓ Time structured merge tree === TSM tree
    - ✓ Efficient indexing for high transactional data, e.g. log data
    - ✓ Write-ahead log (WAL) is used to
      - ✓ write an optimized storage format and
      - ✓ map index files to memory
    - ✓ TSM files contain data points
    - ✓ TSM files have 4 sections:
      - ✓ Header,
      - ✓ Blocks,
      - ✓ Index,
      - ✓ Footer



5 – Installation on Ubuntu

hochschule hof
University of Applied Sciences

1. Install InfluxDB

```
apt-get update && apt-get install influxdb
```

2. Start InfluxDB

```
systemctl start influxdb
```

3. If necessary, enable InfluxDB at boot time

```
systemctl stOP influxdb
systemctl enable influxdb
systemctl unmask influxdb
systemctl start influxdb
```

4. Check InfluxDB configuration

```
influxd config
```

5. Check Influx client

```
influx
```

6 – Configuration Basics

hochschule hof
University of Applied Sciences

1. Show InfluxDB configuration

```
influxd config
```

2. InfluxDB configuration file

```
/etc/influxdb/influxdb.conf
```

3. Disable reporting

```
reporting-disabled = true
```

4. Service start, status, stop, restart

```
service influxdb {start|status|stop|restart}
systemctl {start|status|stop|restart} influxdb
```

5. Check Service using CLI influx

```
you@se-psi40:~$ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
```

7 – Production Deployment

hochschule hof
University of Applied Sciences

1. High Availability
   - InfluxDB supports high availability cluster
   - Each instance has a meta node (system information)
   - Each instance has a data node (actual data)
   - Replication and distribution over networks

2. Backups
   - Critical for production mode
   - Backups are instance snapshots
   - Backups are always full backups

3. Security
   - Enable password authentication
   - Manage users and their permissions restrictively
   - Enable HTTPS in production environments

8 – InfluxDB Query Language

hochschule hof
University of Applied Sciences

1. Start the CLI with human readable time format

```
influx -precision rfc3339
```

2. Check for available databases

```
show databases
```

3. Switch to database

```
use _internal
```

4. Show measurements

```
show measurements
```

5. Select data from measurement runtime

```
SELECT * from runtime limit 3;
SELECT HeapAlloc, HeapIdle, HeapInUse from runtime limit 3;
```

9 – Exercise

hochschule hof
University of Applied Sciences

Create your first InfluxDB database.

1. Login to host master-se using PuTTY or an equivalent terminal software.
2. Take a copy of the provided market data file.

```
cp /tmp/data/ticker_data.txt .
```

3. Edit the file and change the database name to be yourUserName (2 locations!).

```
nano ticker_data.txt
```

4. Load the data from file ticker_data.txt.

```
influx -import -path=ticker_data.txt -database=yourUserName -precision=s
```

5. Verify your import.

```
show databases;
use yourUserName;
show measurements;
select * from tickers;
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement

The **`SELECT`** statement is used to retrieve data points from one or more measurements.
The InfluxDB query syntax is as follows:

```
SELECT <field_key>[,<field_key>,<tag_key>]
FROM <measurement_name>[,<measurement_name>]
WHERE <conditional_expression> [(AND|OR) <conditional_expression> [...]]
```

- Specify the identifier's type

```
SELECT "<field_key>"::field,"<tag_key>"::tag ...
```

- Fully qualified measurements

```
FROM <database_name>.<retention_policy_name>.<measurement_name>
```

- Measurement with default RP

```
FROM <database_name>..<measurement_name>
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement Examples

```
1. select * from tickers;
2. select ticker, company, close, volume from tickers;
3. SELECT ticker::tag,company::field,volume::field,close::field
   FROM tickers limit 3;
4. SELECT *::field FROM tickers limit 3;
5. SELECT ticker, company, volume, close
   FROM yourUserName.autogen.tickers limit 3;
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

WHERE-Clause-Operators

WHERE-Operator	Operation
=	equal to
<>	not equal to
!=	not equal to
>	greater than
>=	greater than or equal
<	less than
<=	less than or equal

Use single quotes for comparing tag-values, because those are of type string:

```
tag_key <operator> 'tag_value'
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement Examples 2

1. Return all tickers when high is larger than 174.6 and ticker is AAPL:

```
SELECT * FROM tickers WHERE ticker='AAPL' AND high > 174.6 limit 3;
```

2. Basic arithmetic to find out the date when the ticker is JPM and high price is more than the low price by 0.2%:

```
SELECT * FROM tickers WHERE ticker='JPM' AND high>low*1.002 limit 5;
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement with GROUP BY

InfluxDB supports group by. The GROUP BY statement is often used with aggregate functions (COUNT, MEAN, MAX, MIN, SUM) to group the result set by one or more tag keys.

The query syntax is as follows:

```
SELECT_clause
FROM_clause
[WHERE_clause]
GROUP BY [* | <tag_key>[,<tag_key]]
```

GROUP BY	Description
`GROUP BY *`	Groups by all tags
`GROUP BY <tag_key>`	Groups by a specific tag
`GROUP BY <tag_key>,<tag_key>`	Groups by multiple specific tags

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

Aggregate Functions

InfluxDB supports several aggregate functions:

Aggregate Function	Description
`COUNT()`	Counts select results.
`DISTINCT()`	Returns only distinct events.
`INTEGRAL()`	Computes the area under the curve.
`MEAN()`	Computes the mean or average.
`MEDIAN()`	Returns the median within the 50% quantile.
`MODE()`	Computes the mode or value that occurs most often.
`SPREAD()`	MAX()-MIN() of a specified column.
`STDDEV()`	Computes the standard deviation.
`SUM()`	Computes the sum in a specified column.

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

Selector Functions

Selector Function	Description
`BOTTOM(field_key,N)`	Returns the smallest N field values associated with the field key.
`FIRST(field_key)`	Returns the oldest field value associated with the field key.
`LAST(field_key)`	Returns the newest field value associated with the field key.
`MAX(field_key)`	Returns the greatest field value associated with the field key.
`MIN(field_key)`	Returns the lowest field value associated with the field key.
`PERCENTILE(field_key,N)`	Returns the Nth percentile field value associated with the field key.
`SAMPLE(field_key,N)`	Returns N randomly selected field values associated with the field key.
`TOP(field_key,N)`	Returns the greatest N field values associated with the field key.

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement with GROUP BY time interval

The **SELECT** statement is used to retrieve data points from one or more measurements. The InfluxDB query syntax is as follows:

```
SELECT <function>(<field_key>)
FROM_clause WHERE <time_range>
GROUP BY time(<time_interval>),[tag_key] [fill(<fill_option>)]
```

Syntax	Description
`time_range`	Range of timestamp field, time > past time and < future time.
`time_interval`	Duration literal, group query result into the duration time group.
`fill (<fill_option>)`	When time intervals have no data, it can change to fill the value.

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-Statement Exercises 3

1. Find all tickers' maximum volumes, minimum volumes, and median volumes during 10-19, November 27, 2017:

```
SELECT MAX(volume), MIN(volume), MEDIAN(volume)
FROM tickers
WHERE time <= '2017-11-27T19:00:00.000000000Z'
     AND time >= '2017-11-27T10:00:00.000000000Z'
GROUP BY ticker;
```

2. What is the change in the SELECT statement above if we use `GROUP BY *` ?

3. Read the documentation for all aggregate and selector functions.
4. What are the formulas behind the aggregate functions INTEGRAL(), MEDIAN(), MEAN()?
5. What is the meaning behind the selector function PERCENTILE()?
6. Create examples for all aggregate and selector functions.

10 – InfluxDB --- DML

SELECT-Statement Exercises 3 continued

7. Find stock trade volumes' mean value during 10-19, November 27, 2017, and group the results into 60-minute intervals and ticker tag key:

```
SELECT mean(volume)
FROM tickers
WHERE time <= '2017-11-27T19:00:00.000000000Z'
     AND time >= '2017-11-27T10:00:00.000000000Z'
GROUP BY time(60m), ticker;
```

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-INTO-Statement

InfluxDB SELECT statement supports the INTO clause. You can copy from one measurement to another or create a new database for the measurement:

```
SELECT_clause INTO <measurement_name>
FROM_clause [WHERE_clause] [GROUP_BY_clause]
```

Syntax	Description
`INTO <measurement_name>`	Copy data into a specific measurement.
`INTO <database_name>.<retention_policy_name>.<measurement_name>`	Copy data into a fully qualified measurement.
`INTO <database_name>..<measurement name>`	Copy data into a specified measurement with a default RP.
`INTO <database_name>.<retention_policy_name>.:MEASUREMENT FROM /<regular_expression>/`	Copy data into a specified database with RP and using regular expression match from the clause.

10 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

SELECT-INTO-Statement Exercises 4

Copy the aggregate mean of all of the field values into a different database and create a new measurement:

```
USE yourDb
SELECT mean(*)
INTO yourDb2.autogen.yourMeasurement
FROM /.*/
WHERE time >= '2017-11-27T00:00:00.000000000Z'
        AND time <= '2017-11-27T23:59:59.000000000Z'
GROUP BY time(60m);
```

Verifikation

```
USE yourDb2
SELECT * from yourMeasurement

name: yourMeasurement
time mean_close mean_high mean_low mean_open mean_volume
---- ---------- --------- -------- --------- -----------
2017-11-27T05:00:00Z 435.3618644067796 435.43688926553676 ...
```

11 – InfluxDB --- DML

hochschule hof
University of Applied Sciences

Query Pagination

InfluxDB supports powerful query pagination using offset and limit.

By getting a total result count, we can easily get the total page number using the following formula:

Total page = (total result count / page size)

If the page size is n, then the initial page displays records starting from 0 to (n-1). When you click on Next, the next page will display records from n to (2n-1), and so on.

The InfluxDB query syntax is as follows:

```
SELECT_clause [INTO_clause]
FROM_clause [WHERE_clause]
[GROUP_BY_clause][ORDER_BY_clause]
LIMIT_clause OFFSET <N> [SLIMIT_clause]
```

Check it out!

```
SELECT ticker,company,volume,close FROM tickers LIMIT 10 OFFSET 100
```

12 – Interaction via Rest API

hochschule hof
University of Applied Sciences

InfluxDB supports a powerful HTTP API that runs by default on port 8086. This allows applications store and pull data through this HTTP API.

InfluxDB has four end points for interaction with,

/ping
/debug/requests
/query
/write

1. PING-Endpoint
   - Command: **`curl -sl --noproxy "*" -I http://localhost:8086/ping`**
   - Take care of the noproxy parameter!
   - Result:

```
HTTP/1.1 204 No Content
Content-Type: application/json
Request-Id: caea1f69-2803-11eb-806c-005056a67b90
X-Influxdb-Build: OSS
X-Influxdb-Version: 1.8.9
X-Request-Id: caea1f69-2803-11eb-806c-005056a67b90
Date: Mon, 22 Oct 2021 08:43:51 GMT
```

12 – Interaction via Rest API

hochschule hof
University of Applied Sciences

2. Debug-Request-Endpoint
   - Open a browser window and load this URL:

     ```
     http://se1-nrd.daas.hof-university.de:8086/debug/requests?seconds=60
     ```

   - In a PuTTY window send the following command several times:

     ```
     curl --noproxy "*"
          -G 'http://localhost:8086/query?db=yourDbName'
          --data-urlencode
          'q=SELECT * FROM yourMeasurement limit 5'
     ```

   - Result after 60 sec in JSON raw data format:

     ```
     {
      "127.0.0.1": {"writes":0,"queries":18}
     }
     ```

12 – Interaction via Rest API

hochschule hof
University of Applied Sciences

3. Query-Endpoint
   - The query request is used to retrieve data points, handle retention points, users, and manage the database.
   - The query request supports the HTTP GET and POST methods.
   - SELECT and SHOW commands can be run under the GET method.
   - The POST method handles database DDL operations:
     ALTER, CREATE, DELETE, DROP, GRANT, KILL, and REVOKE.
   - Several queries may be run separated by semikolons.
   - Examples (use single or double quotes)

```
curl --noproxy "*" -i -XPOST http://localhost:8086/query
                  --data-urlencode "q=CREATE DATABASE testdb"

curl --noproxy "*" -G 'http://localhost:8086/query'
                  --data-urlencode 'q=SHOW DATABASES'

curl --noproxy "*" -i -XPOST http://localhost:8086/query
                  --data-urlencode "q=DROP DATABASE testdb"
```

12 – Interaction via Rest API

hochschule hof
University of Applied Sciences

4. Write-Endpoint
   - The write endpoint is used to write data into a database using the HTTP method POST.
   - The upcoming example shows the upload of two points into database testdb and measurement weather_report.

     The green and blue blocks are separated by a new line.
     The two points are surrounded by single red quotes.

```
curl --noproxy "*" -i -XPOST 'http://localhost:8086/write?db=testdb'
--data-binary 'weather_report,city=NYC
temp=44.1,windchill=42.5,humidity="58%",pressure="30.59
in",winddir="variable",windspeed="3.5 mph" 1511361000000000000
weather_report,city=NYC
temp=41.3,windchill=39.2,humidity="62%",pressure="30.65
in",winddir="NE",windspeed="Calm" 1511361060000000000'
```

13 – InfluxDB API Client

hochschule hof
University of Applied Sciences

- InfluxDB supports many popular programming languages as an application client to interact with the database.
- Write application client code to load and get the data from InfluxDB.
- These are the activities you have to carry out

1. Connect InfluxDB.
2. Create or use the database.
3. Create the points.
4. Write the points to measurement.
5. Query the saved results.

13 – InfluxDB with Python API Client

hochschule hof
University of Applied Sciences

- Installation (Ubuntu 20.04 LTS)
  - Installation: **`apt-get install python3`**
  - Verification: **`python3 –V`**
  - Installation: **`apt-get install python3-pip`**
  - Verification: **`pip3 –V`**
  - Installation: **`pip3 install InfluxDB`**
  - Make sure that the NO_PROXY environment variable is set!
    Linux Bash: **`export NO_PROXY=localhost`**

- Write now your first InfluxDB-Python client!
  - See the upcoming Getting Started Example.

- Check out the Python InfluxDB API documentation:
  - https://influxdb-python.readthedocs.io/en/latest/api-documentation.html

13 – InfluxDB with Python API Client

hochschule hof
University of Applied Sciences

- Getting Started Example
  - Open the python3 shell and check it line by line!
  - Check moodle for the getting started example file!

```
from influxdb import InfluxDBClient
json_body = [ {"measurement": "cpu_load_short",
               "tags": {  "host": "server01", "region": "us-west" },
               "time": "2009-11-10T23:00:00Z",
               "fields": { "value": 0.64 }}]

client = InfluxDBClient('localhost')
client.create_database('example')
client.get_list_database()

client.switch_database('example')

client.write_points(json_body)
result = client.query('select value from cpu_load_short; ')
print("Result: {0}".format(result))

client.drop_database('example')
```

14 – Retention Policy

hochschule hof
University of Applied Sciences

Retention Policy (RP) specifies how much data should be available at any given time.

RP is defined as follows:

- **Duration**: This is the time for how long InfluxDB keeps data
- **Replication factor**: This is the number of copies of the data that are stored in the cluster
- **Shard group duration**: This is the time range that is covered by shard groups
- **Shard groups**: This is the logical container for shards and shards contains actual data

Default RP is autogen.

- If RP is not defined when we create the database, the default autogen RP will be applied.
- RP autogen has an infinity duration, where the replication factor is 1 and the shard group duration is 7.
- When RP duration is more than 6 months, InfluxDB will set the shard Group duration to 7 days by default.

Syntax:

CREATE RETENTION POLICY <retention_policy_name> ON <database_name> DURATION <duration> REPLICATION <n> [SHARD DURATION <duration>] [DEFAULT]

14 – Retention Policy

hochschule hof
University of Applied Sciences

Syntax

```
CREATE RETENTION POLICY <retention_policy_name>
        ON <database_name>
        DURATION <duration>
        REPLICATION <n>
        [SHARD DURATION <duration>]
        [DEFAULT]

ALTER RETENTION POLICY <retention_policy_name>
        ON <database_name>
        DURATION <duration>
        REPLICATION <n>
        SHARD DURATION <duration>
        DEFAULT
```

Take care: Dropping a retention policy will permanently delete all measurements and data stored in the retention policy!

```
DROP RETENTION POLICY <retention_policy_name>
        ON <database_name>
```

15 – InfluxDB Operations

hochschule hof
University of Applied Sciences

What are InfluxDB operations ?

- Back up and Restore data,
- Retention Policies (RP) (see previous slides),
- Monitoring InfluxDB,
- Clustering InfluxDB, and
- High Availability (HA)

15 – InfluxDB Operations

hochschule hof
University of Applied Sciences

Backup

- It is critical not to backup your data !
- In case problem occurs, such as system crashes and hardware failures, you need to be able to restore the data !
- InfluxDB provides a variety of backup and restore strategies.
- Metastore Backup contains system information:
  ```
  influxd backup <path-to-backup>
  ```
- Database Backup: Each database needs to be backed up separately:
  ```
  influxd backup -database <mydatabase> <path-to-backup>
  ```
- Remote Backup
  ```
  influxd backup -host <remote-node-IP>:8088 <path-to-backup>
  ```

15 – InfluxDB Operations

hochschule hof
University of Applied Sciences

Restore

- If ever you need to restore a backup, the restore command is the following:

```
influxd restore [ -metadir | -datadir ]
          <path-to-meta-or-data-directory>
          <path-to-backup>
```

15 – InfluxDB Operations

hochschule hof
University of Applied Sciences

Clustering & High Availability

- InfluxData (company developed InfluxDB) has InfluxEnterprise products, which provide highly scalable clusters.
- Cluster Installation & Requirements
  - set up meta nodes and
  - set up data nodes
  - each meta node runs on its own server
  - at least three meta nodes are needed in a cluster
  - best practices is to deploy each data node on a dedicated server
  - a minimum of two data nodes must be in a cluster for high availability and redundancy and to join data nodes to the cluster

15 – InfluxDB Operations

hochschule hof
University of Applied Sciences

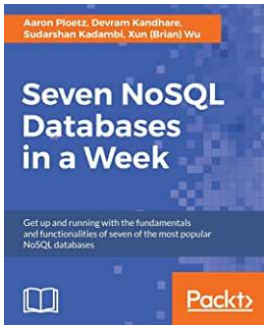
Monitoring Tools

- InfluxDB is a time-series database, which naturally supports real-time event-based data, but there are many tools that can easily integrate with InfluxDB.
- **Kapacitor** is a very useful tool used to process real-time data and generate warning base on tick scripts. It supports both batch and stream by querying the InfluxDB database and defining alert rules and output alert.
- **Chronograf** is another of InfluxData's open source tools for web visualization. It has tick components to monitor data for the InfluxDB database and easily generates visualization alerts in the dashboard.
- **Chronograf offers a UI for Kapacitor**, which is useful for monitoring infrastructure, databases, and so on.

Literature

hochschule hof
University of Applied Sciences

- Seven NoSQL Databases in a Week
  Aaron Ploetz, Devram Kandhare (2018)
  ISBN-13: 978-1787288867



- Go to via VPN to Hof Universities eBook Collections
  Choose platform: Learning O'Reilly
  Login to O'Reilly using Hof University SSO
  https://learning.oreilly.com/library/view/seven-nosql-databases/