Sensing and Actuation Networks and Systems [2022-2023]

Assignment 06 – Using a time-series database for IoT data storage

Introduction

This assignment introduces a time series database to save data from sensors.

Objectives

Students successfully concluding this work should be able to:

- Understand the use of time series databases to store IoT data
- Use InfluxDB for data storage together with Python

Support Material

- InfluxDB Cloud free plan (https://www.influxdata.com/) – Registration required
- Python3 (installed in every student machine)
- Python files: writer.py and reader.py

Requirements

- Internet access
- Knowledge of Python programming!
- InfluxDB Cloud free plan (https://www.influxdata.com/) – Registration required
- Python3 (installed in every student machine)

Note

All the examples and exercises in this assignment can be done by students both at the University and at home.

Background

In an IoT scenario the data produced is normally associated with a time stamp. This is the case when collecting performance metrics, monitoring devices, collecting sensor data or events.

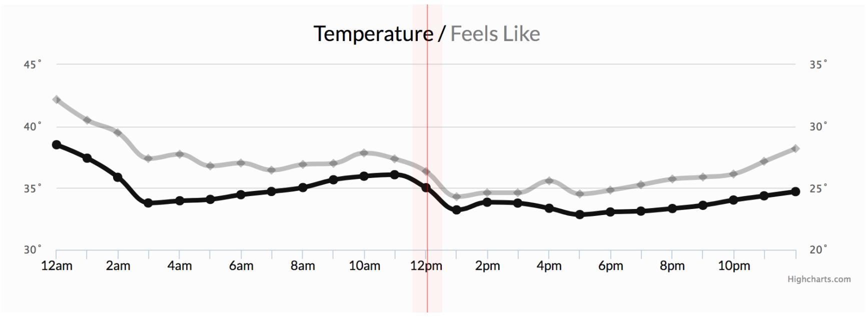


Fig. 1 – Example of time-series data (image source: https://www.influxdata.com/what-is-time-series-data/)

To be useful data has to be analysed. The process that takes data from sources to the final analysis tools is commonly known as ETL: Extract, Transform, Load.

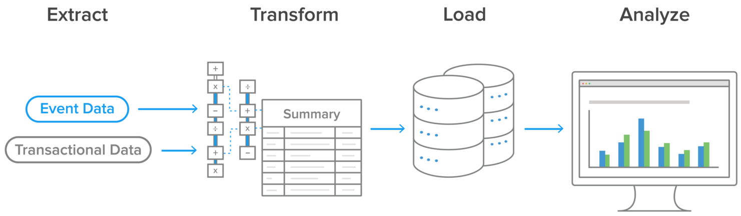


Fig. 2 – ETL process (image source: https://www.stitchdata.com/etldatabase/etl-process/)

The storage and retrieval of IoT time-series data requires high-performance solutions for dealing with the vast number of tuples. This scenario is not suitable for transactional databases that are optimized to manage transactions and involve the update of data, supporting the relationship between different tables. On the contrary, time-series databases are built from scratch for time-series analysis, where data is written and not to be changed anymore (it can be deleted).

In brief, a time series database (TSDB) is a database optimized for time-stamped (time series) data and for measuring change over time. In this assignment InfluxDB will be used.

InfluxDB is an open-source TSDB, designed for fast, high-availability storage and retrieval of time series data in fields such as operations monitoring, application metrics, IoT sensor data, and real-time analytics.

For some key concepts about InfluxDB time-series database read:
https://docs.influxdata.com/influxdb/v1.8/concepts/key_concepts/

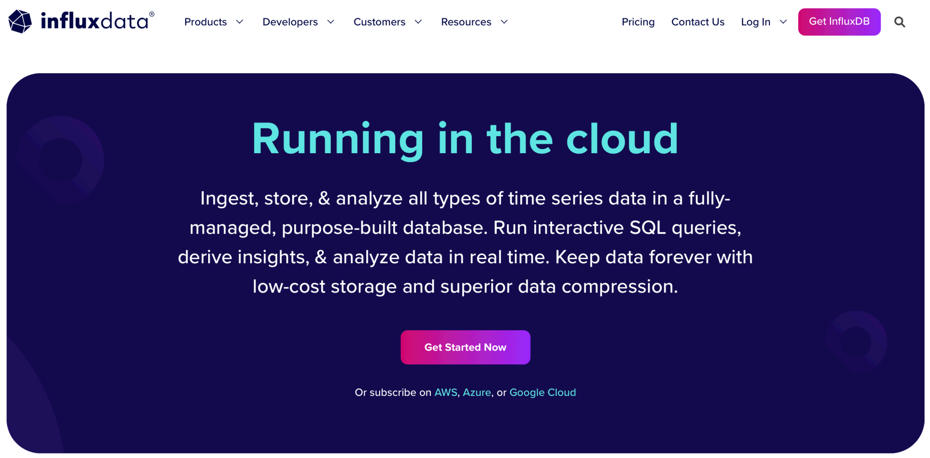
InfluxDB Python Client Library

The InfluxDB Python client library is available directly from PyPI (https://pypi.org/project/influxdb-client/) for easy installs with pip:
To install: `pip install influxdb-client`

Set-up

Creating an InfluxDB Cloud account

To avoid having to install InfluxDB in our own machines, InfluxDB Cloud will be used (https://www.influxdata.com/products/influxdb-cloud/).



Running in the cloud

Ingest, store, & analyze all types of time series data in a fully-managed, purpose-built database. Run interactive SQL queries, derive insights, & analyze data in real time. Keep data forever with low-cost storage and superior data compression.

Fig. 3 – InfluxDB set-up begin

Click on “Get Started Now” and sign up for a free plan - the free plan has some restrictions in performance, only supports 2 databases, and only keeps the data for 30 days.

When asked for a *company name* and *project name*, any name can be used. Choose it wisely. As the *Storage Provider* “aws” in “EU Frankfurt” can be used.

You are now ready to start.

Setting up Client Libraries and test connection

Login into the InfluxDB Cloud account created in the previous step. In the left menu choose “Sources” and the Python Client Library. Following the steps you will learn how to install the influxdb-client modules in your local system to be able to interface with InfluxDB (Python 3 is needed). Follow all the steps and test your connection.

Notes:

Create the Bucket “SRSA” and specify it to delete data after 30 days (the maximum time available)

Copy the given Token to a text file for later use. To avoid saving the token as an environmental variable, copy it directly to the example given on the site:

- Instead of:
  token = os.environ.get("INFLUXDB_TOKEN")
- Use:
  token = ”[write here your token]”

Example

Send and retrieve data to/from your online Influx time-series database

In this example a scenario that includes 2 devices, each with a temperature and humidity sensor, will be created. The values produced by both devices will be simulated using a Python program.

Writer

The sensor values will be randomly generated and sent to a InfluxDB database by a Python program given with this assignment (writer.py).

Analyse the program, change the code to use the specifics of your InfluxDB Cloud account and database, and run it in a terminal window.

To verify that the data is reaching the database, using your InfluxDB Cloud account, access the left menu and choose “Data Explorer”. See the values that are being inserted using the following command in :

```
SELECT * FROM "[Measurement]"
```

Reader

A Python program reader.py will access the InfluxDB database and every 5 seconds will retrieve the number of tuples created in last hour, the last 3 rows submitted and present the average, maximum and minimum values of the last 30 seconds.

Analyse the program, change the code to use the specifics of your InfluxDB Cloud account and database, and run it (in a different terminal window than the writer.py).

Exercise

Using the previous example as reference simulate a scenario where 3 devices send sensor data to an InfluxDB database. Consider the following guidelines:

- Create a separate bucket for this exercise (if needed remove the previous one)
- Device1 and Device 2 generate sensor data of pressure, temperature and humidity every 3 seconds.
- Device3 measures wind speed from 2 different sensors every 2 seconds
- Every 6 seconds read the top 3 results saved in the database

Use one measurement for Device1 and Device2, and other for Device3.

Bibliography

- What is time series data?
    - https://www.influxdata.com/what-is-time-series-data/
- Get started with InfluxDB Cloud
    - https://docs.influxdata.com/influxdb/cloud/get-started/
- Key concepts about InfluxDB
    - https://docs.influxdata.com/influxdb/v1.8/concepts/key_concepts/
- InfluxDB Python Client API
    - https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/#write-data-to-influxdb-with-python
- InfluxDB Python Client WriteAPI
    - https://www.influxdata.com/blog/influxdb-python-client-library-deep-dive-writeapi/
- InfluxDB line protocol
    - https://docs.influxdata.com/influxdb/cloud/reference/syntax/line-protocol/
- Getting started with Python and InfluxDB
    - https://thenewstack.io/getting-started-with-python-and-influxdb/
- InfluxDB SQL Queries with Python:
    - https://www.influxdata.com/blog/influxdb-sql-queries-python/