Time series db analyze

 Could you do it all in MySQL or Postgres? Possibly, but you'd have to write a lot of code to add the functionality many of these databases already provide.

Time series data has certain qualities to it. It's usually write once and requires a balance of high read and write performance.

Only free and open source time series databases and their features have been compared.

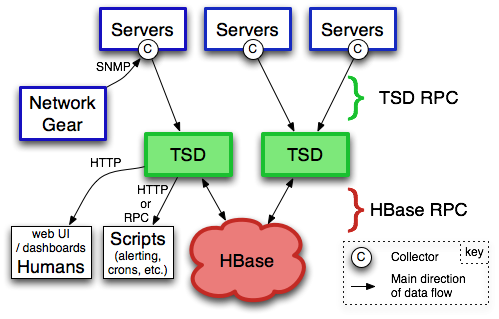
**InfluxDB** is an open-source [time series database](https://en.wikipedia.org/wiki/Time_series_database) developed by InfluxData. It is written in [Go](https://en.wikipedia.org/wiki/Go_(programming_language)) and optimized for fast, high-availability storage and retrieval of [time series](https://en.wikipedia.org/wiki/Time_series) data in fields such as operations monitoring, application metrics, [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_Things) sensor data, and real-time analytics. It also has support for processing data from [Graphite](https://en.wikipedia.org/wiki/Graphite_(software)).[[2]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-Turnbull2014-2)

A **time series database (TSDB)** is a software system that is optimized for handling [time series data](https://en.wikipedia.org/wiki/Time_series), arrays of numbers indexed by time (a datetime or a datetime range). In some fields these [*time series*](https://en.wikipedia.org/wiki/Time_series) are called profiles, curves, or traces. A time series of stock prices might be called a price curve. A time series of energy consumption might be called a load profile. A log of temperature values over time might be called a temperature trace.

A time series database server (TSDS) is a software system that is optimized for handling time series data. A log of temperature values over time might be called a temperature trace. Despite the disparate names, many of the same mathematical operations, queries, or database transactions are useful for analysing all of them. And the implementation of a database that can correctly, reliably, and efficiently implement these operations must be specialized for time-series data. We can name property as metric.

KairosDB是一个OpenTSDB的fork，不过是基于Cassandra存储的。由于Cassandra的行比HBase宽，所以KairosDB的Cassandra的默认行大小为3星期，而OpenTSDB的HBase则为1小时。

OpenTSDB consists of a **Time Series Daemon (TSD)** as well as set of command line utilities. Interaction with OpenTSDB is primarily achieved by running one or more of the TSDs. Each TSD uses the open source database[HBase](http://hbase.org/) to store and retrieve time-series data. Users of the TSD never need to access HBase directly. You can communicate with the TSD via a simple telnet-style protocol, an HTTP API or a simple built-in GUI.



Influxdb

InfluxDB is a time series database so it makes sense to start with what is at the root of everything we do: time. In the data above there’s a column called time - all data in InfluxDB have that column. time stores timestamps, and the *timestamp* shows the date and time, in [**RFC3339**](https://www.ietf.org/rfc/rfc3339.txt) UTC, associated with particular data.

InfluxDB can handle hundreds of thousands of data points per second. Working with that much data over a long period of time can create storage concerns. A natural solution is to downsample the data; keep the high precision raw data for only a limited time, and store the lower precision, summarized data for much longer or forever.

InfluxDB offers two features - Continuous Queries (CQ) and Retention Policies (RP) - that automate the process of downsampling data and expiring old data. This guide describes a practical use case for CQs and RPs and covers how to set up those features in InfluxDB.

1, introduce the 3 different kinds of db

2. compare the 3 kinds

3. show an example of the most suitable one I believe