InfluxDB

**What is InfluxDB(link wiki:**

<https://en.wikipedia.org/wiki/InfluxDB>)

**InfluxDB** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [time series database](https://en.wikipedia.org/wiki/Time_series_database) (TSDB) developed by the company InfluxData. It is written in the [Go programming language](https://en.wikipedia.org/wiki/Go_(programming_language)) for 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)).

## History[[edit](https://en.wikipedia.org/w/index.php?title=InfluxDB&action=edit&section=1)]

[Y Combinator](https://en.wikipedia.org/wiki/Y_Combinator_(company))-backed company Errplane began developing InfluxDB as an open-source project in late 2013 for performance monitoring and alerting.[[2]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-2) Errplane raised an $8.1M Series A financing led by [Mayfield Fund](https://en.wikipedia.org/wiki/Mayfield_Fund) and Trinity Ventures in November 2014.[[3]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-3) In late 2015, Errplane officially changed its name to InfluxData Inc. InfluxData raised Series B round of funding of $16 million in September 2016.[[4]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-4) In February 2018, InfluxData closed a $35 million Series C round of funding led by Sapphire Ventures.[[5]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-5) Another round of $60 million was disclosed in 2019.[[6]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-6)

## Technical overview[[edit](https://en.wikipedia.org/w/index.php?title=InfluxDB&action=edit&section=2)]

InfluxDB has no external dependencies[[7]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-Duffy2015-7) and provides an SQL-like language, listening on port 8086,[[8]](https://en.wikipedia.org/wiki/InfluxDB#cite_note-8) with built-in time-centric functions for querying a data structure composed of measurements, series, and points. Each point consists of several key-value pairs called the fieldset and a timestamp. When grouped together by a set of key-value pairs called the tagset, these define a series. Finally, series are grouped together by a string identifier to form a measurement.

Values can be 64-bit integers, 64-bit floating points, strings, and booleans. Points are indexed by their time and tagset. Retention policies are defined on a measurement and control how data is downsampled and deleted. Continuous Queries run periodically, storing results in a target measurement.

## Why use InfluxDB(link: <https://www.influxdata.com/products/influxdb>/

## )

#### **Faster time to awesome**

InfluxDB has everything you need from a time series platform in a single binary – a multi-tenanted time series database, UI and dashboarding tools, background processing and monitoring agent. All this makes deployment and setup a breeze and easier to secure.

#### **Deep insights and analytics**

[Flux](https://www.influxdata.com/products/flux/) is a fourth-generation programming language designed for data scripting, ETL, monitoring and alerting. As a functional language, you can structure queries and separate common logic into functions and libraries that are easily shared and help speed development. Flux can also be used to enrich your time series data with other SQL data stores (Postgres, Microsoft SQL Server, SQLite, and SAP Hana) along with cloud-based data stores (Google Bigtable, Amazon Athena, and Snowflake). Enriching time series data provides context that can provide further insights into your data.

#### **Optimized for developer productivity**

Everything in InfluxDB — ingest, query, storage and visualization — is now accessible in a unified API. This enables faster time to awesome for developers because everything in the platform can now be programmatically accessed and controlled. This is combined with a powerful set of client libraries across 10 languages (like Go, Java, PHP and Python) and a set of InfluxDB command line tools helps developers develop in a way that is most familiar to them.

#### **Start from the UI**

InfluxDB features a best-in-class UI that includes a Data Explorer, dashboarding tools, and a script editor. Use the Data Explorer to quickly browse through the metric and event data you collected and apply common transformations. The Dashboarding tool comes with a handy list of visualizations that help you to see insights from your data faster. And finally, use the script editor to quickly learn Flux with easily accessible examples, auto-completion and real-time syntax checking.

#### **Easy-to-build, easy-to-share templates**

[InfluxDB Templates](https://www.influxdata.com/products/influxdb-templates/) — a new set of tools that includes a packager and a set of pre-made monitoring solutions — allow you to share your monitoring expertise with coworkers and other community members around the world. The InfluxDB Templates gallery features available templates covering some of the most popular tools, applications, and protocols. These templates can also be checked in as code, fitting in with your continuous integration and deployment pipelines to make deploying (and more importantly rolling back) changes painless.

Sample tutorial:

With InfluxDB open source (OSS) [installed](https://docs.influxdata.com/influxdb/v1.8/introduction/installation), you’re ready to start doing some awesome things. In this section we’ll use the influx [command line interface](https://docs.influxdata.com/influxdb/v1.8/tools/shell/) (CLI), which is included in all InfluxDB packages and is a lightweight and simple way to interact with the database. The CLI communicates with InfluxDB directly by making requests to the InfluxDB API over port 8086 by default.

## [Creating a database](https://docs.influxdata.com/influxdb/v1.8/introduction/get-started/#creating-a-database)

If you’ve installed InfluxDB locally, the influx command should be available via the command line. Executing influx will start the CLI and automatically connect to the local InfluxDB instance (assuming you have already started the server with service influxdb start or by running influxd directly). The output should look like this:

* The InfluxDB API runs on port 8086 by default. Therefore, influx will connect to port 8086 and localhost by default. If you need to alter these defaults, run influx --help.
* The [-precision argument](https://docs.influxdata.com/influxdb/v1.8/tools/shell/#influx-arguments) specifies the format/precision of any returned timestamps. In the example above, rfc3339 tells InfluxDB to return timestamps in [RFC3339 format](https://www.ietf.org/rfc/rfc3339.txt) (YYYY-MM-DDTHH:MM:SS.nnnnnnnnnZ).

The command line is now ready to take input in the form of the Influx Query Language (a.k.a InfluxQL) statements. To exit the InfluxQL shell, type exit and hit return.

A fresh install of InfluxDB has no databases (apart from the system \_internal), so creating one is our first task. You can create a database with the CREATE DATABASE <db-name> InfluxQL statement, where <db-name> is the name of the database you wish to create. Names of databases can contain any unicode character as long as the string is double-quoted. Names can also be left unquoted if they contain *only* ASCII letters, digits, or underscores and do not begin with a digit.

Throughout this guide, we’ll use the database name mydb:

Refer this link for more(<https://docs.influxdata.com/influxdb/v1.8/introduction/get-started>

)

# **InfluxDB 1.8 documentation**

InfluxDB is a [time series database](https://www.influxdata.com/time-series-database/) designed to handle high write and query loads. It is an integral component of the [TICK stack](https://influxdata.com/time-series-platform/). InfluxDB is meant to be used as a backing store for any use case involving large amounts of timestamped data, including DevOps monitoring, application metrics, IoT sensor data, and real-time analytics.

## [Key features](https://docs.influxdata.com/influxdb/v1.8/#key-features)

Here are some of the features that InfluxDB currently supports that make it a great choice for working with time series data.

* Custom high performance datastore written specifically for time series data. The TSM engine allows for high ingest speed and data compression
* Written entirely in Go. It compiles into a single binary with no external dependencies.
* Simple, high performing write and query HTTP APIs.
* Plugins support for other data ingestion protocols such as Graphite, collectd, and OpenTSDB.
* Expressive SQL-like query language tailored to easily query aggregated data.
* Tags allow series to be indexed for fast and efficient queries.
* Retention policies efficiently auto-expire stale data.
* Continuous queries automatically compute aggregate data to make frequent queries more efficient.