Monitoring SQL Server with a free open-source solution

Giovanni Luisotto

Who am I

Giovanni Luisotto

SQL Server DBA
I've been working with SQL Server since 2016
- as BI developer and consultant
- as DBA consultant

Agenda

- Common questions and needs
- The TIG stack
- Telegraf
  - Overview
  - Configure, test and run
  - The SQL Server plugin
- InfluxDB
  - Configure and run
  - InfluxDB basics
- Grafana
  - Install
  - Overview

Common questions and needs

- What happened last night at 01:00AM? *Track and troubleshoot server activities*
- Is the system performing as usual? *Baselining, compare current vs historical*
- Is the server oversized/undersized? *Monitor server resource consumption*
- How fast are my database growing? *Monitor db size*

The TIG stack

Telegraf
Data Gathering

InfluxDB
Time Series Database

Grafana
Analytics platform

The TIG stack - Telegraf

Telegraf is a plugin driven data collector agent written in GO

- One standalone executable
- Lightweight
- Configured using external files
- Great out of the box capability
  - 180+ inputs
  - ~30 outputs
- Open source, contribute and customize it as you need
- Multi-platform (Windows|Linux|OSX|Docker)

The TIG stack - InfluxDB

InfluxDB is a columnar database built to efficiently manage time series data, this kind of DB is usually called Time Series DB (TSDB)

- Non-relational columnar database (schema free)
- Data retention and downsampling
- High data compression
- Great read and write performance
- Poor update and delete performance
- CLI and API (GUI on a separate tool)
- Queried using InfluxQL or Flux
- Multi-platform (Windows|Linux|OSX|Docker)

The TIG stack - Grafana

Grafana is an analytics platform for querying, visualizing and alerting on metrics

- User management & security
- CLI, API and GUI
- Alert rules and notification channels
- Multi-platform (Windows|Linux|OSX|Docker)
- Great community contribution
  - 100+ custom datasources, panels and more
  - Hundreds of ready-to-use dashboards

The TIG stack - additional components

Chronograf

- Administrative GUI for InfluxDB (some commands not available)
- Extremely useful for data exploration and quick query building
- Can be used for dashboarding

Kapacitor

- Data processing engine for InfluxDB
- Has his own scripting language called “tick script”
- Powerful as alerting tool, can define complex rules
- Several notification channel (SMTP/Telegram/Slack)

Telegraf

Telegraf - overview

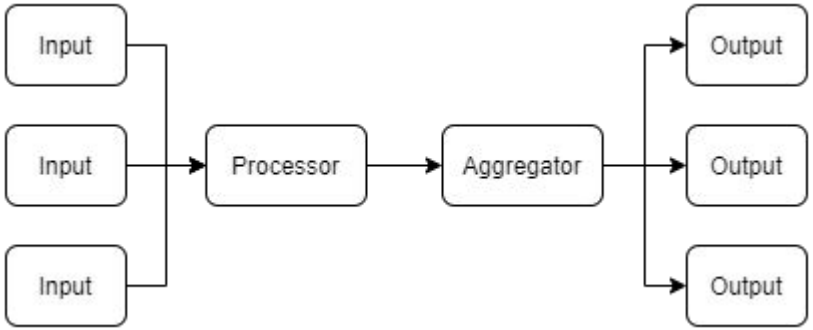
Telegraf is a plugin-driven gathering agent

- Several plugins types
  - Input - fetch data
  - Processor - alter and enrich data
  - Aggregator - aggregate fetched data
  - Output - write data
- Configured using TOML files
- Easy to run as a service

Telegraf - overview

With a defined frequency Telegraf:

1. Fetches data
2. Execute processors
3. Execute aggregators
4. Writes the data



In case of output failures the data are saved into the buffer and written the next data flush.
When the buffer fills, the oldest points are dropped.

Telegraf - configuration sample

Telegraf uses TOML files (Tom's Obvious, Minimal Language), a minimal configuration file format

- Easy to read and use
- Basic data types
- Maps to a hash table

Every plugin has his own configuration section and settings.

- Each plugin can be used multiple times
- The [agent] sections contains the global settings

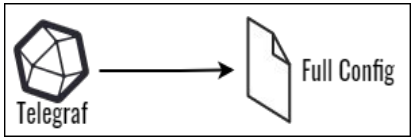
```
[agent]
  interval = "10s"

[[inputs.sqlserver]]
  servers = [
   'Server=QDLP03\SQL2019;app name=telegraf;log=1;',
  ]
  query_version = 2
  # azuredb = false
  # include_query = []
  exclude_query = [ 'Schedulers' , 'SqlRequests' ]

[[outputs.influxdb]]
  urls = ["http://127.0.0.1:8086"]
  database = "myDatabase"
  username = "telegraf"
  password = "telegraf"
```

Telegraf - providing configuration files

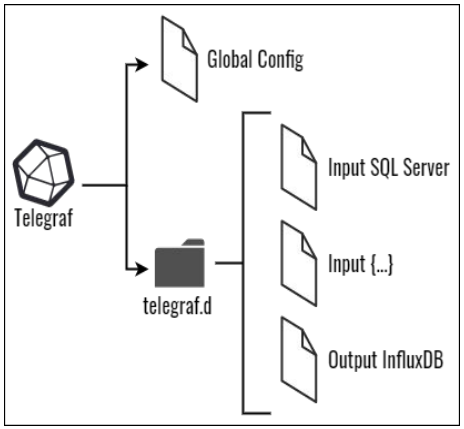
Single File



- The whole configuration resides in a single file

```
Telegraf.exe --config 'PathToConfFile'
```

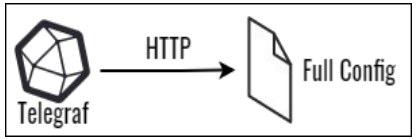
Single file + files folder



- Configuration from a single file + conf files in a specified folder

```
Telegraf.exe --config 'PathToConfFile'
--config-directory 'PathToDir'
```

HTTP GET request



- The whole configuration is fetched by an HTTP call
- Useful for centralizing configuration data

```
Telegraf.exe --config "http://__"
```

Telegraf - run as a service

Telegraf can natively register itself as a Windows service and set some basics settings

- Register/unregister itself
- Set service “name” and “display name”
- Start/stop its service

```
## Register the service with default settings
.\telegraf.exe --config ‘C:\temp\telegraf.conf’ --install
## Register the service with a specific name (useful for multiple Telegraf services)
.\telegraf.exe --config ‘C:\temp\telegraf.conf’ --install --service-name "telegraf_demo" --service-display-name "Telegraf Demo"
## Uninstall a specific Telegraf service
.\telegraf.exe --uninstall --service-name "telegraf_demo"
```

Note: always use the full paths to configuration files, otherwise error 1067 will occur when starting the service

Telegraf - SQL Server plugin

- Supported engine editions
  - Azure SQL DB
  - Azure Managed Instance
  - SQL Server 2008 SP3 and newer
- Supports Windows and SQL authentication
- Collects meaningful metrics
  - Perf counters
  - Wait stats
  - Database IO
  - CPU
  - ... and more

Telegraf - SQL Server plugin - configuration

Parameters

- *servers* - list of connection strings
  - Several parameters available (docs here)
- *query_version* - stick with v2, v1 is deprecated
- *include_query* - list of queries to include, empty means all
- *exclude_query* - list of queries to exclude, has priority over include list
- *azuredb* - enables additional queries for Azure SQL DB

```
[[inputs.sqlserver]]
  servers = [
    'Server=QDLP03\SQL2019;app name=telegraf;log=1;',
    'Server=QDLP03\SQL2016;app name=telegraf;log=1;',
  ]
  query_version = 2
  # azuredb = false
  # include_query = []
  exclude_query = [ 'Schedulers' , 'SqlRequests' ]
  ## Possible queries:
  ## - PerformanceCounters - WaitStatsCategorized
  ## - DatabaseIO - ServerProperties - MemoryClerk
  ## - Schedulers - SqlRequests - VolumeSpace - Cpu
```

Telegraf - SQL Server plugin - authentication & permissions

SQL Server authentication

- Requires user and password in the connection string
- Specified with `User Id=user;Password=password` parameters
- Values can be hidden using environment variables

Windows authentication

- Does not require any parameter
- The user executing Telegraf will be used log-in

SQL Server permissions

- VIEW SERVER STATE
- VIEW ANY DEFINITION

Telegraf - SQL Server plugin - errors & fixes

The SQL Server plugin doesn't have any severe issues, but you can still incur in some errors/warnings

Two problems I've often found are

- Error - [outputs.influxdb] {...} max key length exceeded {...}
- Warning - [inputs.sqlserver] did not complete within its interval

Both can slow or stop the data gathering process, especially when monitoring more than one SQL Server instance in the same input plugin instance

Note: On service start, the first 1-2 read/write often fails with a variety of error regarding connection to SQL Server or InfluxDB, don't worry about it unless the error persist

Telegraf - SQL Server plugin - errors & fixes

```
E! [outputs.influxdb] When writing to {...} max key length exceeded: ___ > 65535
dropped=0; discarding points
```

Reason

- the “statement text” is a tag, part of the key
- The key exceeds its maximum size (not configurable)

Behaviour

- The whole batch writing will fail, valid point kept in buffer
- Once the buffer fills, the oldest points will be dropped

Fix

- Convert the “statement text” to a field
- This also makes the “statement_text” string readable

```
[[processors.converter]]
  [processors.converter.tags]
    Namepass = ["sqlserver_requests"]
    string = ["statement_text"]
```

Telegraf - SQL Server plugin - errors & fixes

```
W! [agent] [inputs.sqlserver] did not complete within its interval
```

Reason

- At least one of the SQL instances did not complete within the defined interval
- Usually caused by network problems

Behaviour

- The whole input sqlserver session will fail
- Data of instances who completed within time will be lost

Fix

- Specify the *"connection timeout"* in the connection string (in seconds)
- Will generate an error in case of timeout
- The other gathered data will be written

```
[[inputs.sqlserver]]
  servers = [
    'Server=QDLP03\SQL2019;app name=telegraf;log=1;connection timeout=5;',
    'Server=QDLP03\SQL2016;app name=telegraf;log=1;connection timeout=5;',
  ]
  {...}
```

InfluxDB

InfluxDB - executables & configuration

InfluxDB comes with several executables and a sample configuration file

- influx.exe → InfluxDB CLI (command line)
- influxd.exe → InfluxDB database executable

Its internal default configuration includes

- Database files are created in the current user folder (ie: C:\Users\gluisotto\.influxdb)
- Database authentication is disabled
- Logs are sent to stdout
- The default port is 8086

InfluxDB uses the following configuration precedence

1. environment variables → 2. configuration file → 3. internal default configuration

InfluxDB - run as a service using NSSM

To run InfluxDB as a service you need to use NSSM (Non-Sucking Service Manager) which allows us to run executables/commands as services on windows.

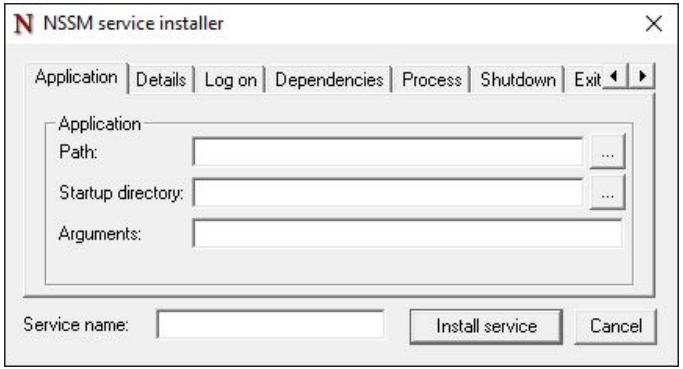
NSSM offers

- command line and minimal GUI
- Advanced service configuration
- Input/Output redirection (useful for logging)

Important notes

- Run it as administrator
- Use only full paths in command argument

```
# run nssm GUI
nssm.exe install
# uninstall a previously registered service
nssm.exe remove __servicename__
```



SQL DB and InfluxDB Comparison

InfluxDB	SQL DB
Database	Database
Retention Policy	-
Measurement	Table
Point	Row
Tag	Column (indexed)
Field	Column (unindexed)

InfluxDB - point structure

<measurement>,<tag set> <field set> <timestamp>

sqlserver_cpu,host=QDLP03,sql_instance=QDLP03:SQL2017 other_process_cpu=6i,sqlserver_process_cpu=0i 1593694804000000000

A point has 4 components

- Measurement - namespace/container of the point
- Tag set - key value pairs, provides the context
- Field set - key value pairs, provides the metrics
- Timestamp - UNIX timestamps with nanosecond precision

A **series** is logical grouping of data defined by shared **measurement**, **tag set**, and **field key**
The **point key** is made by the combination of **timestamp** and **series**

InfluxDB - retention and downsampling

Retention Policies (RP)

- Manages data retention
  - Period, replication and shard size
- A DB can have multiple RP
- The name must be unique per DB

```
CREATE RETENTION POLICY "<rp_name>" ON "<db>" DURATION 1w REPLICATION 1
```

Downsampling data

- Keep high precision data for a limited time
- Aggregate data points for long term storage
- Achieved with **continuous queries** (CQ)
- Requires multiple RP

InfluxDB - continuous queries

A CQ is a recurring query that aggregates and store data points

```
CREATE CONTINUOUS QUERY "sql_cpu" ON "quantumdatis_demo"
BEGIN
  SELECT
    mean("sqlserver_process_cpu") AS "sqlserver_process_cpu"
  INTO "quantumdatis_demo"."long"."sqlserver_cpu_hist"
  FROM "quantumdatis_demo"."standard"."sqlserver_cpu"
  GROUP BY
    time(5m)
    ,*
END
```

- Runs with the frequency specified in the GROUP BY statement
- Aggregates data for the time window specified in the GROUP BY
- Advanced syntax allows to specify different frequency and time window

```
CREATE {...} ON "<db>"
RESAMPLE EVERY <interval> FOR <interval>
BEGIN {...}
```

InfluxDB - security

By default authentication is disabled, it can be enabled by changing the InfluxDB configuration. Once enabled you will be forced to create a db admin user.

InfluxDB has only 2 types of user and a total of 3 permissions

Admin users

- READ and WRITE on all databases
- Access to management commands (SHOW/CREATE/ALTER/DROP of DB, RP, CQ, Measurements, Series etc)

```
CREATE USER <username> WITH PASSWORD '<password>'
WITH ALL PRIVILEGES
```

Non-admin users

- Permission per single database
- READ, WRITE, ALL (READ and WRITE)

```
CREATE USER <username> WITH PASSWORD '<password>'
GRANT READ ON "<database>" TO "<username>"
```

Grafana

Grafana - installation

Grafana for windows comes in two ways:

- A zip folder to manually install it
- And msi installer (recommended)

Once installed you can access grafana using

- From port 3000
- The default user “admin” with password “admin”

The Grafana configuration is stored in files

- Defaults.ini - do not use, gets overridden on update
- Custom.ini - has priority over Defaults.ini, can be a partial config

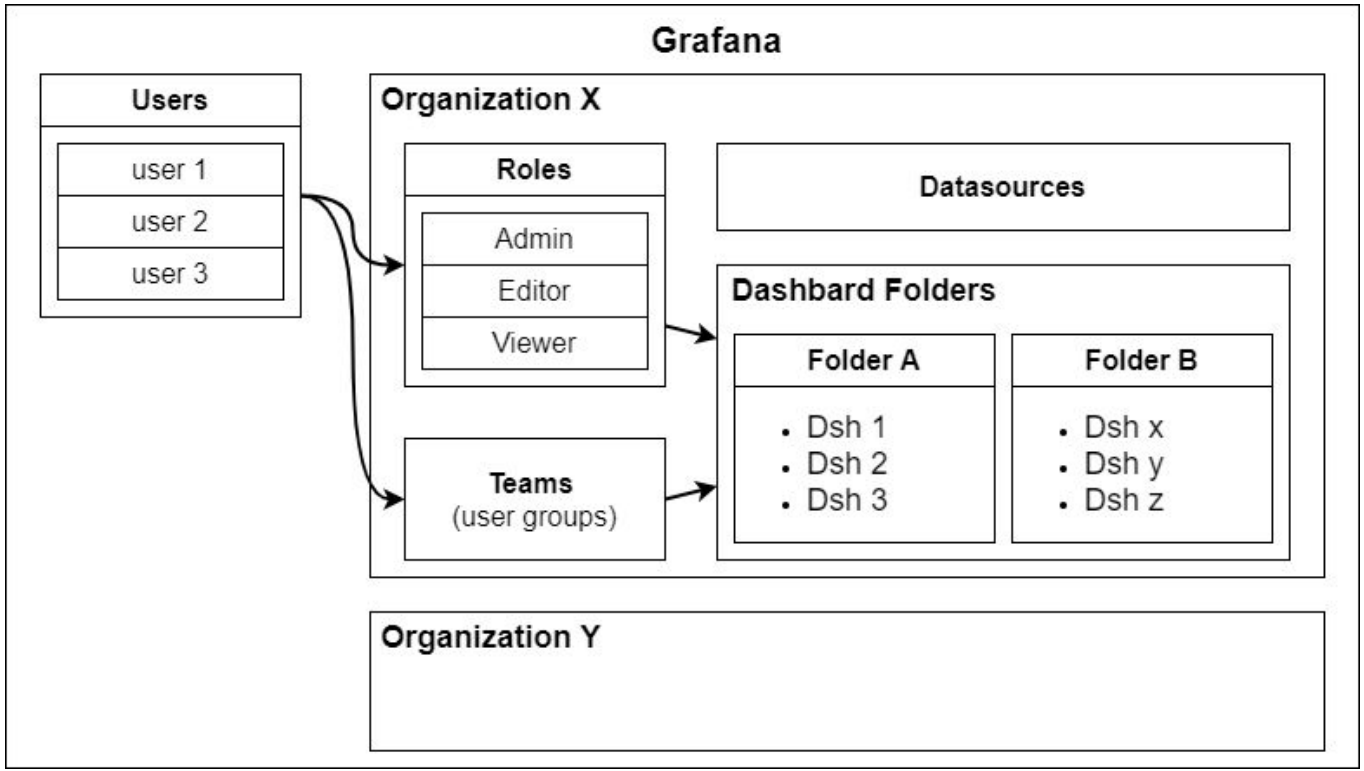
Grafana - structure overview

Server level

- Users
- Organizations

Organization level

- Datasources
- Folders
- Dashboards
- Teams
- Roles (fixed)



Grafana - security

Server level

- All the users are created at Instance level
- User can be mapped to organizations
- Users can be set as Grafana Admin

Grafana Admins

Can manage

- Users
- Organizations

Cannot manage

- Organization contained objects (teams, folders, etc)

To manage all objects, map your user as Grafana admin and organization admin

Organization Level

There 3 fixed permissions type

- Admin
- Edit
- View

Permissions can be set on objects like

- Folders
- Dashboards
- Datasources (only Grafana Enterprise)

Permission can be assigned to

- Roles (default)
- Teams
- Users

Grafana - dashboards & panels features

Dashboards

- Are saved in JSON format
- JSON definition can be viewed and edited
- Versioned in an internal repository
- Can be exported as JSON files
- Can be imported from
  - JSON Files/text
  - Grafana Dashboards site
- Can use multiple data sources
- Define parameters
- Automatic (local) timezone conversion

Panels

- Are visuals and charts
- Can use multiple queries and sources
- Guided or manual query definition
- Copy-paste and view JSON definition
- Excellent unit of measure management
- Wide formatting options
- Additional panels can be downloaded from the community
- Some functions/features are data-source dependant

Grafana demo

Useful resources

Documentation & guides

- Telegraf introduction (link)
- Telegraf plugin list (link)
- Telegraf GitHub repository (link)
- InfluxDB introduction (link)
- Grafana introduction (link)
- Grafana dashboards (link) - community made dashboards
- Grafana playground (link) - samples of Grafana features

Forums

- InfluxData Community - help and suggestions on Telegraf, InfluxDB and more
- Grafana Community - help, suggestions and guides on Grafana

Summary

Why should you use it?

- Complete monitoring solution
- Easy to set-up
- Simple to manage
- Free and open-source
- Multi-platform