

Telegraf — InfluxDB — Grafana

Einsammeln eigener Daten via CSV

Dirk Geschke



Linux User Group Erding

27. Mai 2020

Gliederung

- 1 Überblick
- 2 Telegraf
- 3 InfluxDb
- 4 Grafana

Zusammenspiel

Die drei Einheiten bilden einen Klassiker:

Telegraf sammelt Daten ein und leitet sie weiter

InfluxDB speichert Zeitreihen in Datenbank

Grafana visualisiert die Daten

Besonderheiten von Telegraf

- optimiert für Zeitreihen
- geringer Speicherverbrauch
- sammelt zeitgesteuert Daten
- kann Daten umwandeln oder aggregieren
- leitet diese an Datenbank weiter
- über 200 Plugins
- Open Source
- geschrieben in Go

Installation von Telegraf

Dies gilt für Debian 10: GPG Key für Repository hinzufügen:

```
# curl -sL https://repos.influxdata.com/influxdb.key  
| \  
apt-key add -  
OK
```

Hinzufügen des Repositorys:

```
# echo "deb https://repos.influxdata.com/debian \  
buster stable">/etc/apt/sources.list.d/influxdata.list
```

Aktualisieren und Installieren:

```
# apt update; apt install telegraf
```

Output von Telegraf für InfluxDB

Wichtig sind hier diese Einstellungen:

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

Input von Telegraf für CSV

Wichtig sind hier diese Einstellungen:

```
[[inputs.exec]]
  commands = [ "/usr/local/bin/collect.pl" ]
  timeout = "5s"
  csv_header_row_count = 1
  csv_column_types=["string","int","int","int","int"]
  csv_delimiter = ","
  data_format = "csv"
```

Das Programm gibt einfach Speicherdaten als CSV aus:

```
#!/usr/bin/perl
use strict;
my $sw; my $fr; my $bu; my $ca;
open (FD, "vmstat |");
my $l=<FD>; $l=<FD>; $l=<FD>;
close(FD);
print "measurement,Swap,Free,Buffer,Cache\n";
if ($l =~ /^s+(.*)/) {$l=$1;}
($sw,$fr,$bu,$ca)=(split(/\s+/, $l))[2,3,4,5];
print "Speicher,$sw,$fr,$bu,$ca\n";
```

Besonderheiten von InfluxDB

- high performance speziell für Zeitreihen
- CLI, HTTP-API, Client-Bibliotheken, Plugins
- SQL-ähnliche DB-Sprache
- Continues Querys (*down sampling*)
- Retention Policy (*data expiration*)
- Open Source
- geschrieben in Go

Installation von InfluxDB

Die Installation von InfluxDB ist unter Debian sehr einfach:

```
apt install influxdb
```

Das Repository wurde bereits bei der Installation von Telegraf hinzugefügt!

Konfiguration von InfluxDB

Die Anpassungen gelten für Telegraf

- verbinden mit DB:

```
# influx
```

```
Connected to http://localhost:8086 version 1.8.0
```

```
InfluxDB shell version: 1.8.0
```

```
>
```

- erzeugen der Datenbank:

```
> CREATE DATABASE telegraf
```

Konfiguration von InfluxDB

- anzeigen der Datenbanken:

```
> SHOW DATABASES
name: databases
name
----
_internal
telegraf
```

Konfiguration von InfluxDB

- User anlegen:

```
> CREATE USER telegraf WITH PASSWORD 'geheim'
```

- Zugriff erlauben:

```
> GRANT ALL ON telegraf TO telegraf
```

- anzeigen der Benutzer:

```
> SHOW USERS
```

```
user admin
```

```
-----
```

```
telegraf false
```

Konfiguration von InfluxDB

- optionale Datenlöschung

```
> CREATE RETENTION POLICY thirty_days ON  
telegraf DURATION 30d REPLICATION 1 DEFAULT
```

- Retention Policy für Löschung alter Daten
- Dauer 30 Tage
- Voreinstellung für Datenbank `telegraf`

Besonderheiten von Grafana

- Visualisierung von Daten
- anpassbare Dashboards (sehr flexibel!)
- Plugins auch für InfluxDB
- Alarme
- Open Source
- geschrieben in Go

Installation von Grafana

Zuerst wieder den GPG-Key hinzufügen:

```
# curl -sL https://packages.grafana.com/gpg.key | \
apt-key add -
OK
```

dann das Repository:

```
# echo "deb https://packages.grafana.com/oss/deb \
stable main">/etc/apt/sources.list.d/grafana.list
```

und zum Abschluss installieren:

```
# apt update; apt install grafana
```

Abschluss von Grafana

- starten des Programms: `/etc/init.d/grafana start`
- log in via: `http://localhost:3000/`
 username **admin**
 password **admin**
- eigentlich noch **reverse Proxy** davor, z.B. *nginx*
- kann dann auch **TLS**-Absicherung übernehmen
- weitere Konfiguration über **Web-GUI**

Praxis...!