# Stack Telegraf / InfluxDB / Grafana

# **InfluxDB**

## Connexion à la BDD

Tout simplement (si on se trouve sur le serveur qui héberge la base de données et en supposant que le client influx est dans le PATH)

influx -precision rfc3339 -database 'influxdb'

# **Correction d'une SERIE**

#### Contexte:

J'ai des erreurs de mesures<sup>1)</sup> remontées par une de mes sondes de température, que je

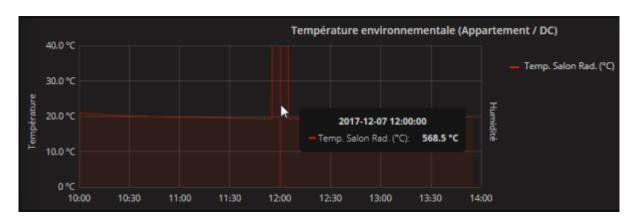
souhaite corriger à la méthode du garagiste Islandais<sup>2)</sup> (Private Joke):



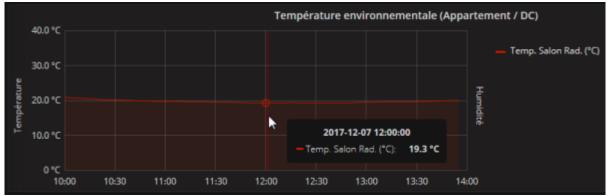
- Nom de la base InfluxDB : influxdb
- La SERIE est stockée dans la MEASUREMENT Temperature\_temper
- Fréquence de collecte : 5 min / 300 sec
- Valeur à corriger :
  - Points de **température > 100°C** (la pièce à vivre n'est pas censée être un



## **Avant** correction:



Après correction :



On commence par se connecter à la base de données qui va bien...

On affiche la structure de la MEASUREMENT

- **▼** Structure toute simple, contenant 1 SERIE avec :
  - 1 tagKey host
  - 1 fieldKey value de type Float

Commençons par évaluer l'ampleur de la tâche à réaliser (et voir si il ne serait pas plus efficace de créer un script...)

Bon ok, 1 point de mesure à corriger, c'est pas si pire...



Pour appliquer la 2ème méthode, qui implique d'utiliser le timestamp au format **epoch** (précision à la nanoseconde), je ne saurais que conseiller aussi de lister les points à corriger avec le format de précision ns

https://wiki.drouard.eu/ Printed on 22:40 13/12/2025

Liste des points de **température** à corriger

Passons à la correction!

En fonction de la nature, criticité de la métrique, 2 possibilités pour corriger l'erreur :

- Méthode 1 
   Supprimer la valeur erronée :
  - Il est possible de corriger l'apparence de la courbe à la volée dans Grafana avec la clause GROUP BY fill (0/null/none/etc.)... c'est fourbe, mais esthétique... inconvénient, ça

masquerait les erreurs de collectes trop récurrentes



- Méthode 2 Maquiller la valeur erronée (Ce n'est qu'une extension de la méthode 1) :
  - Cette stratégie consiste sournoisement à faire une moyenne (par ex.) entre les 2 valeurs entourant le pic puis à inscrire cette valeur à la place de la valeur du pic

On commence par identifier le laps de temps incriminé via une petite requête (±1 période de collecte) :

On supprime la valeur erronée

```
DELETE FROM Temperature_temper WHERE "host" = 'drouard.eu' AND time =
'2017-12-07T11:00:04Z'
```

On vérifie que la mesure à bien été supprimée

```
> SELECT * FROM Temperature_temper WHERE time >= '2017-12-07T10:54:04Z' AND time <= '2017-12-07T11:06:04Z' name: Temperature_temper
```

Last update: 15:01 07/12/2017

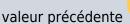
```
value
time
                     host
2017-12-07T10:55:01Z drouard.eu 19.29
2017-12-07T11:05:01Z drouard.eu 19.31
```

La **méthode 1** consiste à en rester là et éventuellement maquiller légèrement la courbe dans Grafana...

Maintenant, on va aller un peu plus loin avec la **méthode 2**.



**Note**: il faut savoir que si l'on insert une valeur avec un **timestamp existant**, alors I' INSERT agira comme un UPDATE (qui d'ailleurs, n'existe pas en tant que tel dans influxDB). Ce qui implique que pour la méthode 2, il n'est pas nécessaire de supprimer la



On va devoir injecter une nouvelle valeur avec comme paramètres :

- **value** = Moyenne des valeurs encadrant
  - $\circ$  Dans cet ex: (19.29 + 19.31) / 2 = **19.30**
- timestamp = Timestamp au format epoch, précision à la nanoseconde
  - Dans cet ex : timestamp = 1512644404000000000



A noter qu'il est possible de convertir une date au format RFC-3339 vers le format epoch avec une précision à la nanoseconde avec la commande date :

```
date -d 2017-12-07T11:00:04Z +%s%N
```

Syntaxe générique d'insertion de donnée dans InfluxDB (Docs de référence sur https://docs.influxdata.com : Writing and exploring data et Line Protocol) :

```
<measurement>[,<tag_key>=<tag_value>[,<tag_key>=<tag_value>]]
<field key>=<field value>[,<field key>=<field value>] [<timestamp>]
```

Ensuite, on peut forger la requête d'insertion soit :

• En mode **CLI** via le client influx

• En mode HTTP API (C'est à la mode... et et c'est pratique aussi) via curl

```
curl -i -XPOST -u USERNAME:PASSWORD
"http://localhost:8086/write?db=db drouard&precision=ns" --data-binary
'Temperature temper, host=drouard.eu value=19.30 15126444040000000000'
```

https://wiki.drouard.eu/ Printed on 22:40 13/12/2025 Enfin, on vérifie la correction

```
> SELECT * FROM Temperature_temper WHERE time >= '2017-12-07T10:54:04Z' AND time <= '2017-12-07T11:06:04Z' name: Temperature_temper time host value ---- 15126441010000000000 drouard.eu 19.29 15126444040000000000 drouard.eu 19.3 15126447010000000000 drouard.eu 19.31
```

1)

Problème corrigé depuis... causé par une concurrence d'accès à la sonde entre Nagios et Telegraf.

Fix It or Remove It:p

From:

https://wiki.drouard.eu/ - Vim Online;)

Permanent link:

https://wiki.drouard.eu/pub\_zone/linux/stack\_tig?rev=1512658870

Last update: 15:01 07/12/2017

