

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



Work in Progress



Contexte :

J'ai des erreurs de mesures¹⁾ remontées par une de mes sondes de température, que je souhaite corriger à la méthode du garagiste Islandais²⁾ (🤪 Private Joke) :



- Nom de la base InfluxDB : `influxdb`
- La SERIE est stockée dans la MEASUREMENT `Temperature_temperhum`
- Fréquence de collecte : 5 min / 300 sec
- Valeurs à corriger :
 - Points de **température** > **100°C** (la pièce à vivre n'est pas censée être un four 😊)
 - Points d'**humidité** > **100%** (100% étant l'humidité de l'eau, je ne vois pas bien ce que pourrait être une humidité de 110% 😄)

On commence par [se connecter à la base de données](#) qui va bien...

On affiche la structure de la **MEASUREMENT**

```
> SHOW SERIES FROM Temperature_temperhum
KEY
---
Temperature_temperhum,host=drouard.eu
```

```
> SHOW TAG KEYS FROM Temperature_temperhum
name: Temperature_temperhum
tagKey
-----
host

> SHOW FIELD KEYS FROM Temperature_temperhum
name: Temperature_temperhum
fieldKey fieldType
-----
VALUE      FLOAT
```

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...)

```
> SELECT COUNT("value") FROM Temperature_temperhum WHERE VALUE > 100;
name: Temperature_temperhum
TIME          COUNT
-----
1970-01-01T00:00:00Z 10

> SELECT COUNT("value") FROM Humidity_temperhum WHERE VALUE > 100;
name: Humidity_temperhum
TIME          COUNT
-----
1970-01-01T00:00:00Z 10
```

Bon ok, une vingtaine de points 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

Liste des points de **température** à corriger

```
> PRECISION rfc3339
> SELECT * FROM Temperature_temperhum WHERE VALUE > 100 LIMIT 20;
name: Temperature_temperhum
TIME          host          VALUE
-----
2017-12-04T14:50:01Z drouard.eu 568.5
2017-12-04T19:25:03Z drouard.eu 568.5
2017-12-04T20:05:01Z drouard.eu 568.5
```

```

2017-12-04T22:15:05Z drouard.eu 568.5
2017-12-04T23:40:01Z drouard.eu 568.5
2017-12-04T23:50:02Z drouard.eu 568.5
2017-12-05T22:10:01Z drouard.eu 568.5
2017-12-06T00:00:05Z drouard.eu 568.5
2017-12-06T00:10:02Z drouard.eu 568.5
2017-12-06T10:45:04Z drouard.eu 568.5

```

```

> PRECISION ns
> SELECT * FROM Temperature_temperhum WHERE VALUE > 100 LIMIT 20;

```

name: Temperature_temperhum

TIME	host	VALUE
-----	-----	-----
1512399001000000000	drouard.eu	568.5
1512415503000000000	drouard.eu	568.5
1512417901000000000	drouard.eu	568.5
1512425705000000000	drouard.eu	568.5
1512430801000000000	drouard.eu	568.5
1512431402000000000	drouard.eu	568.5
1512511801000000000	drouard.eu	568.5
1512518405000000000	drouard.eu	568.5
1512519002000000000	drouard.eu	568.5
1512557104000000000	drouard.eu	568.5

Liste des points d'**humidité** à corriger

```

> PRECISION rfc3339
> SELECT * FROM Humidity_temperhum WHERE VALUE > 100 LIMIT 20;

```

name: Humidity_temperhum

TIME	host	VALUE
-----	-----	-----
2017-12-04T14:50:01Z	drouard.eu	2517
2017-12-04T19:25:03Z	drouard.eu	2517
2017-12-04T20:05:01Z	drouard.eu	2517
2017-12-04T22:15:05Z	drouard.eu	2517
2017-12-04T23:40:01Z	drouard.eu	2517
2017-12-04T23:50:02Z	drouard.eu	2517
2017-12-05T22:10:01Z	drouard.eu	2517
2017-12-06T00:00:05Z	drouard.eu	2517
2017-12-06T00:10:02Z	drouard.eu	2517
2017-12-06T10:45:04Z	drouard.eu	2517

```

> PRECISION ns
> SELECT * FROM Humidity_temperhum WHERE VALUE > 100 LIMIT 20;

```

name: Humidity_temperhum

TIME	host	VALUE
-----	-----	-----
1512399001000000000	drouard.eu	2517
1512415503000000000	drouard.eu	2517
1512417901000000000	drouard.eu	2517
1512425705000000000	drouard.eu	2517

```
1512430801000000000 drouard.eu 2517
1512431402000000000 drouard.eu 2517
1512511801000000000 drouard.eu 2517
1512518405000000000 drouard.eu 2517
1512519002000000000 drouard.eu 2517
1512557104000000000 drouard.eu 2517
```

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... inconvenient, 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 surnoisement 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

Prenons l'exemple de l'erreur du **4 décembre 2017 15h50 UTC+1** (2017-12-04T14:50:01Z). On commence par identifier le laps de temps incriminé via une petite requête :

```
> SELECT * FROM Temperature_temperhum WHERE TIME >= '2017-12-04T14:44:01Z'
AND TIME <= '2017-12-04T14:56:01Z'
name: Temperature_temperhum
TIME          host          VALUE
-----
2017-12-04T14:45:03Z drouard.eu 20.78
2017-12-04T14:50:01Z drouard.eu 568.5
2017-12-04T14:55:04Z drouard.eu 20.72
```

On supprime la valeur erronée

```
DELETE FROM Temperature_temperhum WHERE "host" = 'drouard.eu' AND TIME =
'2017-12-04T14:50:01Z'
```

On vérifie que la mesure à bien été supprimée

```
> SELECT * FROM Temperature_temperhum WHERE TIME >= '2017-12-04T14:44:01Z'
AND TIME <= '2017-12-04T14:56:01Z'
name: Temperature_temperhum
TIME          host          VALUE
-----
2017-12-04T14:45:03Z drouard.eu 20.78
2017-12-04T14:55:04Z drouard.eu 20.72
```

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 l' **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 valeur précédente 😎

On va devoir injecter une nouvelle valeur avec comme paramètres :

- **value** = Moyenne des valeurs encadrant
 - Dans cet ex : $(20.78 + 20.72) / 2 = \mathbf{20.75}$
- **timestamp** = Timestamp au format epoch, précision à la nanoseconde
 - Dans cet ex : `date -d 2017-12-04T14:50:01Z +%s%N = 1512399001000000000`

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`

```
INSERT Temperature_temperhum,host=drouard.eu VALUE=20.75 1512399001000000000
```

- En mode **HTTP API** (C'est à la mode... 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_temperhum,host=drouard.eu value=20.75 1512399001000000000'
```

Enfin, on vérifie la correction

```
> SELECT * FROM Temperature_temperhum WHERE TIME >= '2017-12-04T14:44:01Z'
AND TIME <= '2017-12-04T14:56:01Z'
name: Temperature_temperhum
TIME                host          VALUE
----                -
2017-12-04T14:45:03Z drouard.eu  20.78
2017-12-04T14:50:01Z drouard.eu  20.75
2017-12-04T14:55:04Z drouard.eu  20.72
```

1)
Problème corrigé depuis... causé par une concurrence d'accès à la sonde entre Nagios et Telegraf.

2)
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=1512646841

Last update: **11:40 07/12/2017**

