

# StreamingToDockerInfluxDB

October 12, 2019

## 1 Streaming to InfluxDB 2.0 HTTP server

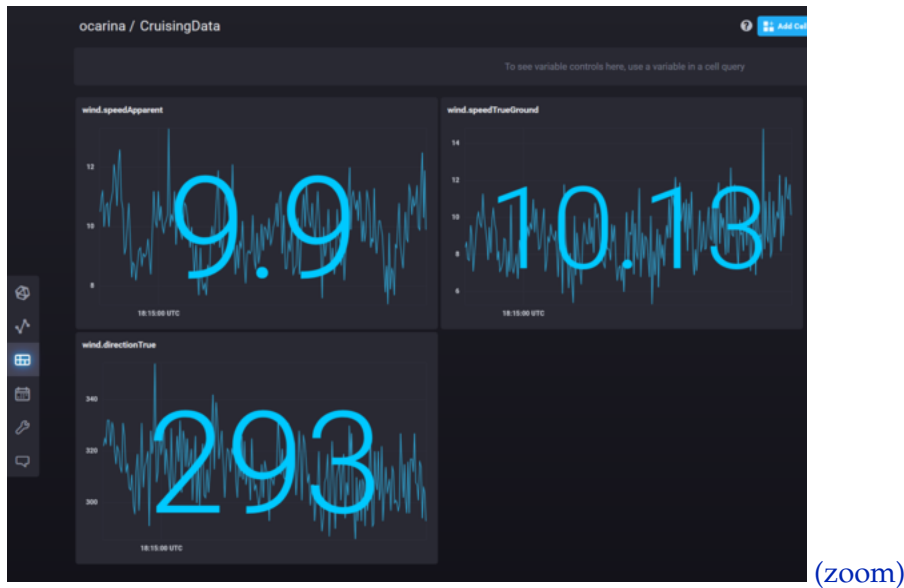
Like explained in *Dashboard-Tactics InfluxDB 2.0 Streamer - User's Guide* ([ipynb](#) | [html](#) | [pdf](#)) the easiest way to get data into an InfluxDB 2.0 server is to create and capture a massive text data file, called Line Protocol file and then import it using InfluxDB 2.0 UI function for the bucket data import.

The static method applied only when back home does not, however allow to use the powerful functions of InfluxDB 2.0 on the time series data, such as filtering, averaging, monitoring, observing the tendency and so forth.

You are familiar with the wind history instrument in Dashboard-Tactics. Well, this is the same but in steroids! By directly connecting Dashboard-Tactics with InfluxDB 2.0 server allows you to have similar functions on **all** data which is received (or produced) byt Dashboard-Tactics!

### 1.1 InfluxDB 2.0 Dashboards

To motivate you, let's skip to the result - what we want is a collection of key data which we want to follow in real-time. In InfluxDB 2.0, they are called *Dashboards*. Basically, you create a query to capture from the incoming time sequenced data stream the data you want to observe in time, **optionally applying filters, algorithms or mathematical functions on the time series data**. Once you are satisfied, you would save your work as a cell in an InfluxDB 2.0 Dashboard. This screenshot gives you an idea what to expect (raw data is shown):



Isn't it curious how *much* the raw values are jumping up and down. Can't we do something for it?

You're already hooked! Let's see how to get things running, first.

## 1.2 Installing InfluxDB 2.0 server on your navigation computer with Docker

If you have just one navigation computer and not a networked cluster of computers in your boat, you need to install the InfluxDB 2.0 server on your navigation computer where the Dashboard-Tactics is sitting together with OpenCPN. It is likely that your computer is Windows, so let's use that as an example, there is not much difference for Linux: we are going to use so called container method.I

### 1.2.1 Install Docker

You need a [Docker desktop](#). It is quite big chunk to download, a decent speed link is needed. [Read more about the installation](#).

Once installed, Docker is not that hungry any more to eat the rest of your disk space; here's mine reporting about the disk consumption, after running some extensive data input tests with InfluxDB 2.0 in a container:

```
> docker system df
```

TYPE	TOTAL	ACTIVE	SIZE
RECLAIMABLE			
Images	1	1	176.5MB
OB (0%)			
Containers	3	1	68.53MB
33.59MB (49%)			
Local Volumes	1	0	36.09MB
36.09MB (100%)			
Build Cache	0	0	0B

### 1.2.2 Install InfluxDB 2.0 container image and launch it

[InfluxDB 2.0 download page](#) give you this simple command to execute with docker:

```
docker pull quay.io/influxdb/influxdb:2.0.0-alpha
```

Yes, currently when this is written it is still alpha (version 14 to be exact) but it is totally usable. Installation is dead easy, launching it will be done by precising on which port InfluxDB 2.0 should serve HTTP:

```
docker run --name influxdb --restart=unless-stopped -p 9999:9999
quay.io/influxdb/influxdb:2.0.0-alpha
```

Open your browser with address <http://localhost:9999> and let you guide through the welcome. Make sure that you don't give too complicated user name and password - this is local and unprotected HTTP anyway and you would be annoyed with too much typing since yes, there is a timeout for a session!

### 1.2.3 Important! Synchronize your clocks

You can imagine that the need for good timekeeping both on client side and on the DB server side is essential. This is not normally an issue but in your computer, probably having Hyper-V virtualization support, any power save state will stop the hardware clock of the container as well. Clock drift will occur.

Write this simple Windows PowerShell script in a file and give it a name like `setClockInfluxDb-Docker.ps1`:

```
# A PowerShell Script to set the InfluxDB container time to the time obtained from the host
# courtesy to
# https://forums.docker.com/t/docker-for-windows-should-resync-vm-time-when-computer-resumes-from-sleep/10000

$influxImgName = "quay.io/influxdb/influxdb:2.0.0-alpha"

$datetime = Get-Date
$dt = $datetime.ToUniversalTime().ToString('yyyy-MM-dd HH:mm:ss')
docker run --net=host --ipc=host --uts=host --pid=host --security-opt=seccomp=unconfined --privileged --rm
sleep 3
```

You can create a shortcut to it on Desktop and modify its Target field like this:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass
-File C:\Users\<YOU>\<YOUR_PATH>\setClockInfluxDbDocker.ps1
```

The script will read your computer's time and set the container time with it. Small script but saves you a lot of headache if you see your container time drifting!

## 1.3 Prepare InfluxDB Out instrument configuration for HTTP

You can find a template `streamout_template_http.json` from `C:\Program Files (x86)\OpenCPN\plugins\dashboard_tactics_pi\data`

Copy (keep the original) it to the run-time data directory C:\ProgramData\opencpn\plugins\dashoard\_tactics\_pi which exists if you have already tried the writing to the InfluxDB 2.0 Line protocol file. It is better to give it a different name than the default one, let's use streamout-http.json.

You can modify the opencpn.ini file and give the new configuration file name to be used like this:

```
[PlugIns/Dashboard/Tactics/Streamout] ConfigFile=streamout-http.json
```

If you do not have that value, you need to start at least once an InfluxDB Out Tactics "instrument".

Modify streamout-http.json file to contain your connection parameters:

```
"org" : "myorg", // HTTP: Influx DB organization name
```

You have selected an organization (boat?) name in InfluxDB 2.0. Type it here in place of myorg - exactly as you defined it in InfluxDB 2.0 UI.

```
"bucket" : "mybucket", // HTTP: Influx DB bucket to write
```

Data is written in buckets of InfluxDB 2.0. Create one and give its exact name here.

```
"token" : "ToLdk3DNs3PqbKNS2hdZMure.....E0eu41E00UWRt8w=="
```

Tokens are unique, per each server, and even per each bucket in it. Go to Tokens-tab in Settings of InfluxDB 2.0 UI and generate one. Copy the new token into clipboard and paste it between string quotes. Do not allow a carriage return, it must be a single string.

Now you are ready to try. But if it does not work? There is debugging information available in OpenCPN log file. But you must turn the debugging on in the configuration file:

```
"verbosity" : 3 // 0=quiet,1=events,2=verbose,3+=debug
```

Don't leave debugging on once the issue is resolved, it can be quite verbose.

## 1.4 Try it out in the InfluxDB 2.0 Data Explorer

Data Explorer allows you to build your queries, which you can save as Dashboard cells, as explained above. But you can use it also to test the incoming data by switching between the graphical and raw data mode.

Set the data period to be last 5 minutes and ask data to be queried every 5 seconds. This gives you pretty good insight of what data is coming in.

And what data is coming in?

You can observe it best by looking at available measurement types, their metadata (tags) and the actual field names. Using the query builder, browse through the available filters so that you end up to select one value, or few values who have something in common and, preferably, similar data range:



(zoom)