Domoticz, influxDB and Garfana (MQTT) for nice graphs

Table of Contents

Description....3
Request:....3
Assumptions:....3
Principle:....3
Variant 1 – Write data direct into influxDB....3
Variant 2 – Write data from Domoticz to influxDB through MQTT....4
Prepare influxDB....4
Send data from domoticz to influxDB....5
Variant 1 - Set Domoticz write data direct into influxDB....5
Variant 2 - Write data from Domoticz to influxDB through MQTT....8
Prepare MQTT broker MOSQUITO....8
Configure DOMOTICZ to public device changes into MQTT....11
Send data from MQTT do influxDB (by python script)....13
Use GRAFANA to draw graphs....13
Více informací:....18
Write by:....18

Description

This document describe how-to create nice graphs from domoticz by program Grafana and auxiliary programs (influxDB, [MQTT + python script]).

Why I do this:
Because Domoticz can draw only basic graphs and older data summarize for save memory. Other-way Grafana can draw nice graphs and influxDB can store data for longer time.

Request:

- Domoticz automation system
- [MQTT server with websocket (best is mossquito) + python with mqtt and influxDB support)
- InfluxDB
- Grafana
- Web browser

Assumptions:

- running domoticz

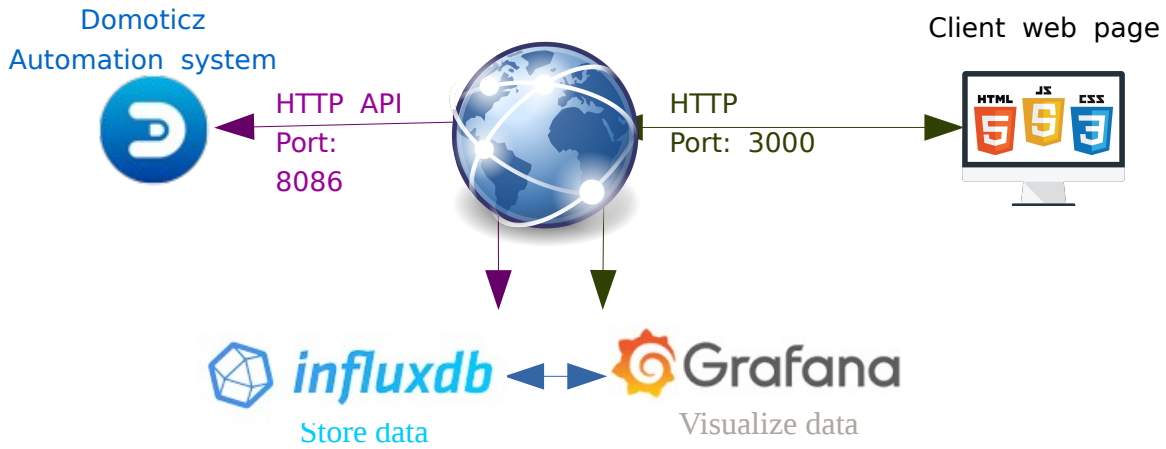
Principle:

In this article are described two variants how to send data from domoticz to influxDB.

Short description used services or programs:

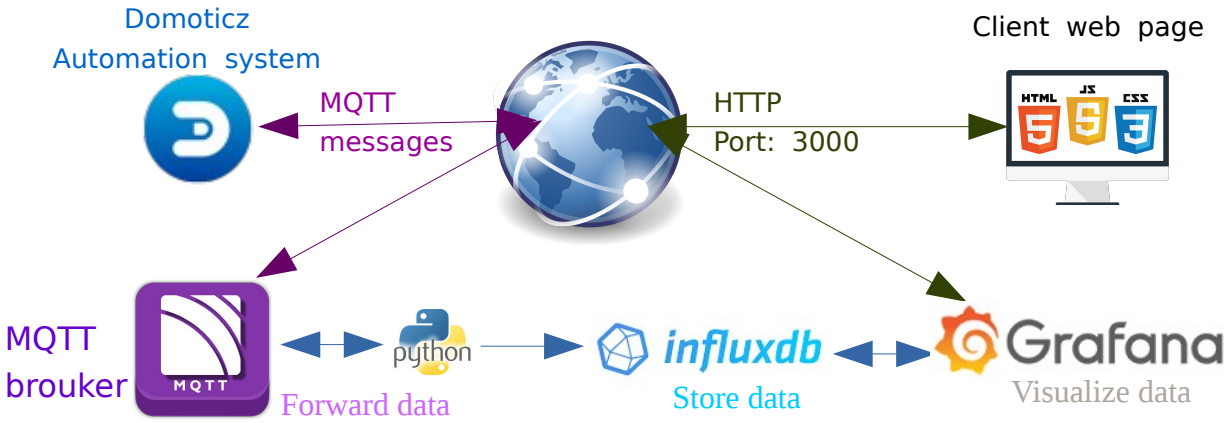
Domoticz – home automation system
(www.domoticz.com)
MQTT – messages server (forwarder) for IOT (internet of things) (only for variant 2)
(https://mosquitto.org)
Python – program language, for run script who sending information from MQTT to InfluxDB (only for variant 2)
InfluxDB – database server, used for store information from domoticz
(https://docs.influxdata.com/influxdb/v1.5/)
Grafana – is web program for visualize data (in this case from influxDB). Draw nice graphs.
(https://garafana.com)

Variant 1 – Write data direct into influxDB



This easer variant.

Variant 2 – Write data from Domoticz to influxDB through MQTT



This variant more difficult but is cleaner when you use communication between Domoticz and MQTT early for another purpose.

In both variants enable view data behind NAT, if you use public influxDB and Grafana server.

Prepare influxDB

Downlload and install

Best guide is on https://portal.influxdata.com/downloads#influxdb.
For ubuntu and debian type on command line:

```
wget https://dl.influxdata.com/influxdb/releases/influxdb_1.5.2_amd64.deb
sudo dpkg -i influxdb_1.5.2_amd64.deb
```

Configure:

If you wish do some change etc. change default ports, than you can do in /etc/influxdb/influxdb.conf.
Default port if 8086 for communication and 8085 for admin web page.

Start deamon:

```
sudo service influxdb start
```

Create new database:

Connect to InfluxDB shell using the commandline

```
influx
Visit https://enterprise.influxdata.com to register for updates, InfluxDB server management, and monitoring.
Connected to http://localhost:8086 version 0.10.0
InfluxDB shell 0.10.0
>
```

Create a database.
For this quick start we'll call the database "*demo*". Run this command inside the InfluxDB shell.

```
> CREATE DATABASE demo
>
```

Only for test:

You can show list databases.

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

>
```

Select database

```
> USE demo
Using database demo
```

write data by influxdb shell

```
INSERT cpu,host=serverA value=0.64
```

or data into influxDB you can also write by standard http request e.g. bash script

```
curl -i -XPOST 'http://localhost:8086/write?db=demo' --data-binary 'cpu,host=serverA value=0.84'
```

select data from influxdb shell

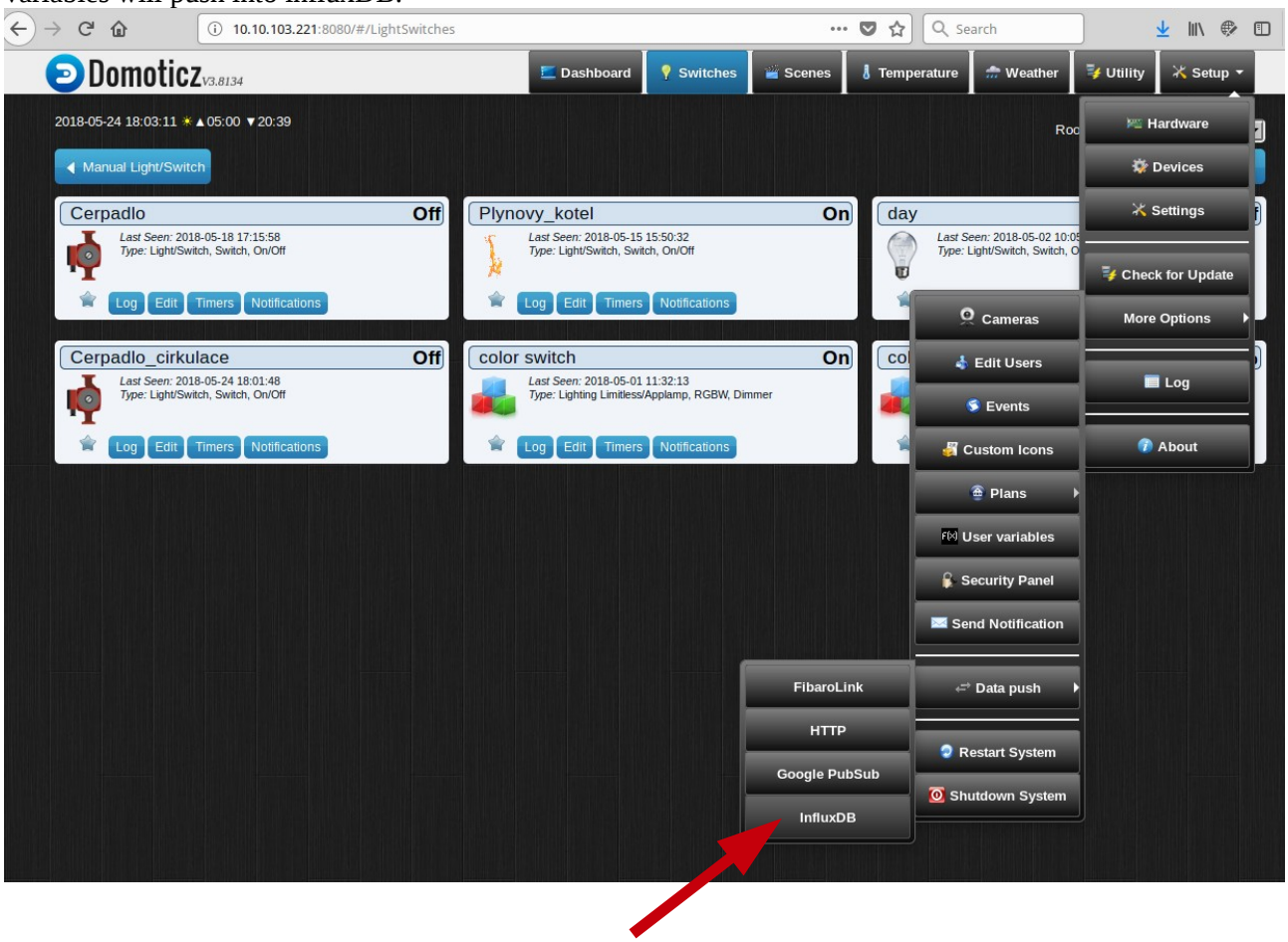
```
> SELECT * FROM cpu
name: cpu
---------
time                    host    value
1527176107506652262             serverA 0.64
1527176252035641431             serverA 2.01
```

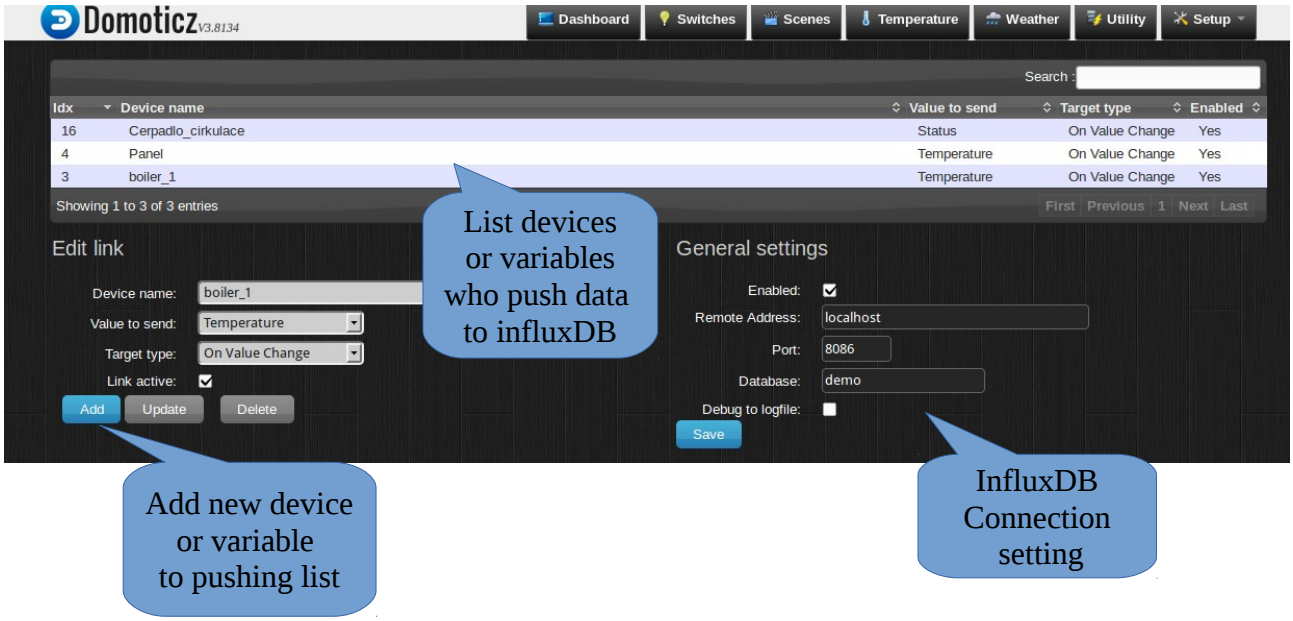
OK database is ready.

Send data from domoticz to influxDB

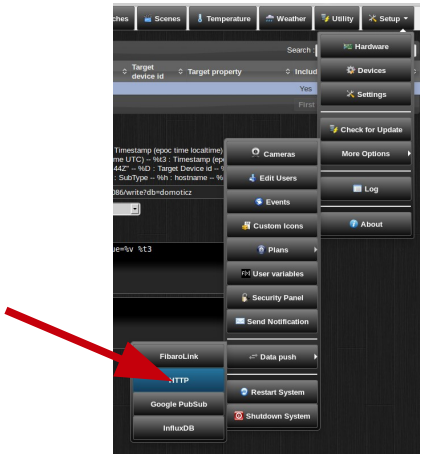
Variant 1 - Set Domoticz write data direct into influxDB

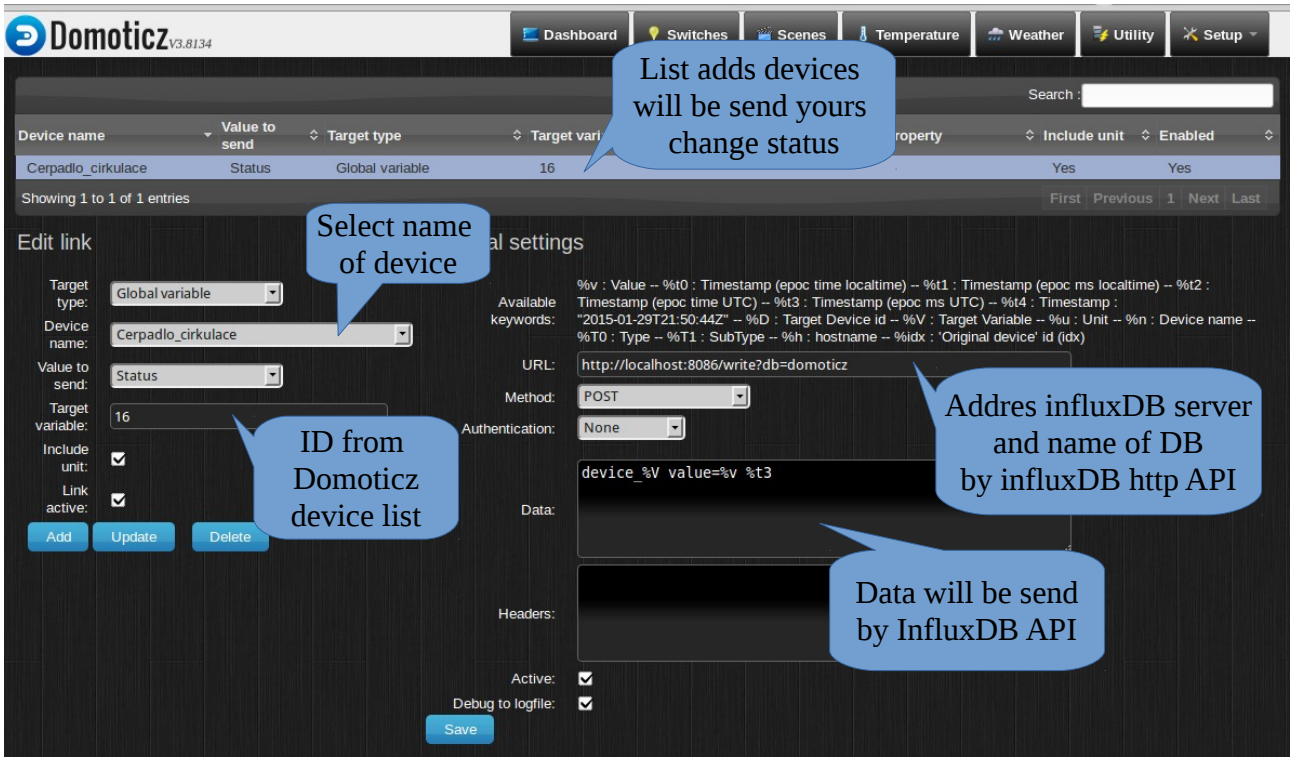
It is very easy. Open menu Hradvare->More Otions->Data pusch->InfluxDB and create list who devices or variables will push into influxDB.





Another possibility howto send is use menu *Hradvare->More Otions->Data pusch->Http* and define http request to push into influxDB by influxDB http api.
This is more difficult, but useful for older versions of domoticz who haven't support for influxDB.





Variant 2 - Write data from Domoticz to influxDB through MQTT

Prepare MQTT broker MOSQUITO

Description:

MQTT is a machine-to-machine messaging protocol, designed to provide lightweight publish/subscribe communication to "Internet of Things" devices. It is commonly used for geo-tracking fleets of vehicles, home automation, environmental sensor networks, and utility-scale data collection.

Mosquitto is a popular MQTT server (or broker, in MQTT parlance) that has great community support and is easy to install and configure.

Install mosquitto:

On ubuntu and debian type on command line:

```
sudo apt-get install mosquitto mosquitto-clients
```

By default, Ubuntu will start the Mosquitto service after install. Let's test the default configuration. We'll use one of the Mosquitto clients we just installed to subscribe to a topic on our broker.

Topics are labels that you publish messages to and subscribe to. They are arranged as a hierarchy, so you could have sensors/outside/temp and sensors/outside/humidity, for example. How you arrange topics is up to you and your needs. Throughout this tutorial we will use a simple test topic to test our configuration changes.

Log in to your server a second time, so you have two terminals side-by-side. In the first terminal, use mosquitto_sub to subscribe to the test topic:

```
mosquitto_sub -h localhost -t test
```

-h is used to specify the hostname of the MQTT server, and -t is the topic name. You'll see no output after hitting ENTER because mosquitto_sub is waiting for messages to arrive. Switch back to your other terminal and publish a message:

```
mosquitto_pub -h localhost -t test -m "hello world"
```

The options for mosquitto_pub are the same as mosquitto_sub, though this time we use the additional -m option to specify our message. Hit ENTER, and you should see hello world pop up in the other terminal. You've sent your first MQTT message!

Configuring MQTT Passwords:

Let's configure Mosquitto to use passwords. Mosquitto includes a utility to generate a special password file called mosquitto_passwd. This command will prompt you to enter a password for the specified username, and place the results in /etc/mosquitto/passwd.

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd sammy
```

Now we'll open up a new configuration file for Mosquitto and tell it to use this password file to require logins for all connections. Should you open an empty file /etc/mosquitto/conf.d/default.conf and paste in the following:

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwd
```

allow_anonymous false will disable all non-authenticated connections, and the password_file line tells Mosquitto where to look for user and password information. Save and exit the file.

Now we need to restart Mosquitto and test our changes.

```
sudo systemctl restart mosquitto
```

Try to publish a message without a password:

```
mosquitto_pub -h localhost -t "test" -m "hello world"
```

The message should be rejected:

```
Output
Connection Refused: not authorised.
Error: The connection was refused.
```

Before we try again with the password, switch to your second terminal window again, and subscribe to the 'test' topic, using the username and password this time:

```
mosquitto_sub -h localhost -t test -u "sammy" -P "password"
```

It should connect and sit, waiting for messages. You can leave this terminal open and connected for the rest of the tutorial, as we'll periodically send it test messages.

Now publish a message with your other terminal, again using the username and password:

```
mosquitto_pub -h localhost -t "test" -m "hello world" -u "sammy" -P "password"
```

The message should go through as in Step 1. We've successfully added password protection to Mosquitto. Unfortunately, we're sending passwords unencrypted over the internet. Fix that next by adding SSL encryption to Mosquitto, but is out of this manual.

Configuring MQTT Over Websockets (Optional)

In order to speak MQTT using JavaScript from within web browsers, the protocol was adapted to work over standard websockets. This is not nessesery for use domoticz, mqtt and grafana, but may be very useful in future for create interactive web pages with domoticz. If you don't need this functionality, you may skip this step.

We need to add one next more listener block to our Mosqiutto config file in /etc/mosquitto/conf.d/default.conf .

```
. . .
listener 8083
protocol websockets
```

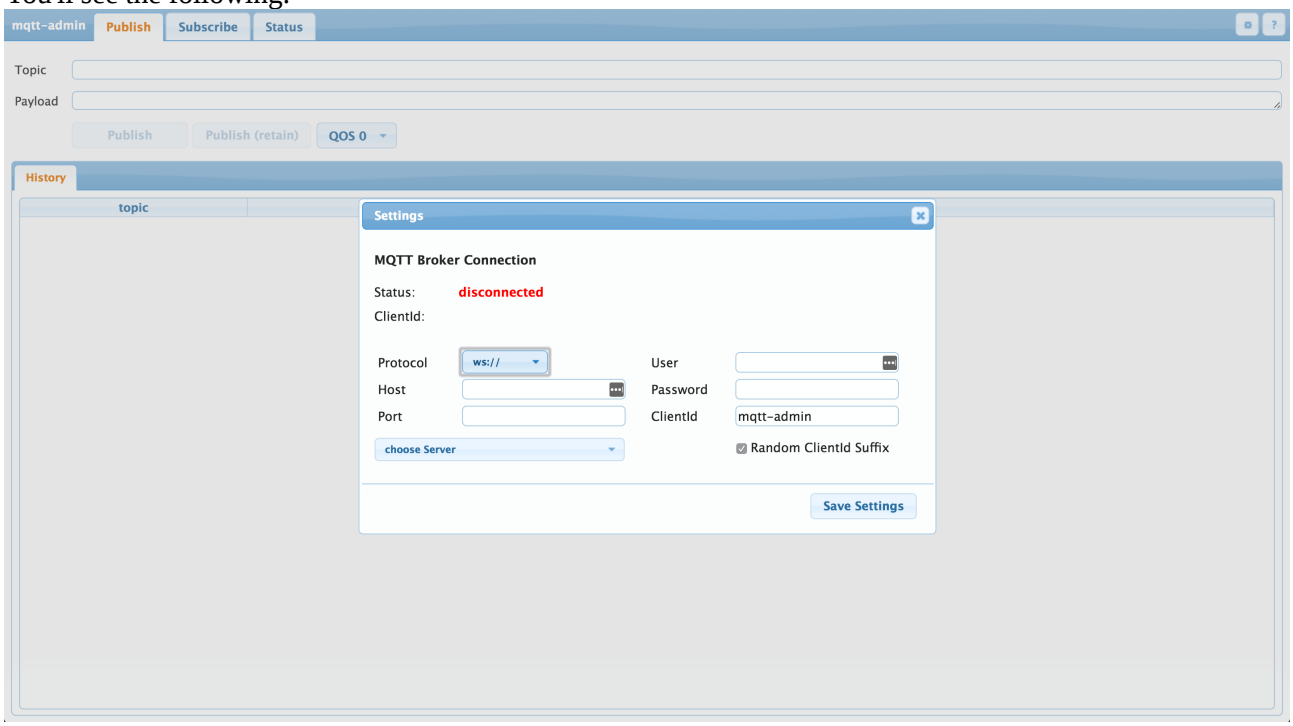
This is mostly the same as the previous block, except for the port number and the protocol websockets line. There is no official standardized port for MQTT over websockets, but 8083 is the

most common.

Save and exit config the file, then restart Mosquitto.

```
sudo systemctl restart mosquitto
```

For simply test is best use browser-based MQTT client as Open mqtt-admin .
You'll see the following:



Fill connection informations to your MQTT broker save settings and you can go test.

Configure DOMOTICZ to public device changes into MQTT

Description:

Domoticz contain direct support for MQTT who send and receivre messages from/to MQTT broker.

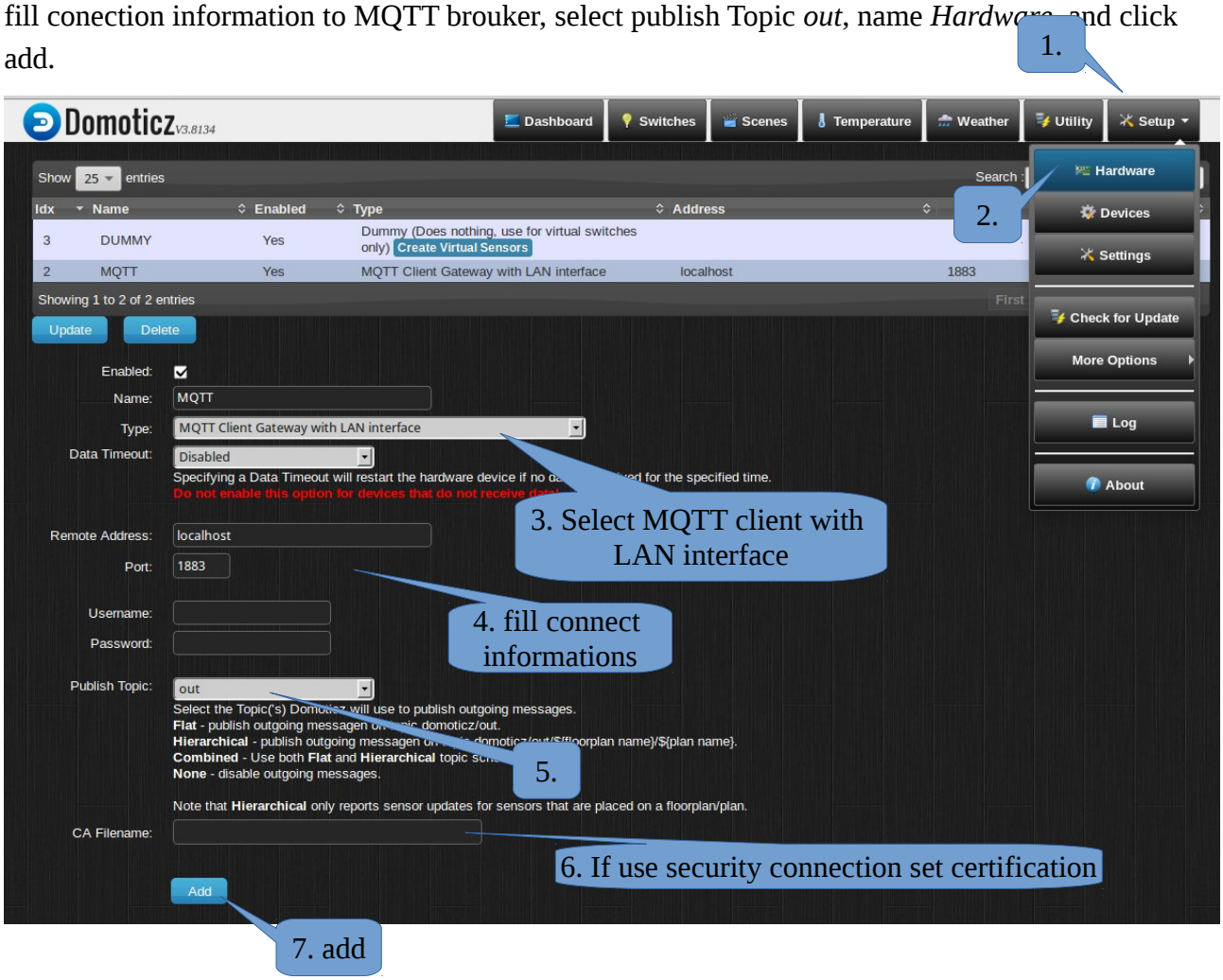
Send mqtt meseges is standard way howto forward data to another aplication. Every messages who domoticz send and accept receive is in JSON format e.g.:

```
{
 "Battery" : 255,
"RSSI" : 12,
"description" : "",
 "dtype" : "Light/Switch",
 "id" : "00014060",
"idx" : 16,
"name" : "Pump_of_circulate",
"nvalue" : 1,
"stype" : "Switch",
"svalue1" : "0",
"switchType" : "On/Off",
"unit" : 1
}
```

more about used JSON format is on https://www.domoticz.com/wiki/MQTT

Configure:

In menu Setting->Hardware find and set type *MQTT Client Gateway with LAN interface*
fill conection information to MQTT brouker, select publish Topic *out*, name *Hardware* and click add.



In (point 5) *Publish Topic* is next slection:

- out – every outdoing message have *domoticz/out* mqqt topic (recomnend)
- / - outdoing messages have topic *domoticz/out/{$forplan name}/{$plan name}* (me not run)
- out + / - combination both of select
- none – domotict not send outdoing mqtt messages

In all case domoticz recivre incoming messages from *domoticz/in* mqtt topic

Deafault MQTT topis for domoticz is:

- domoticz/out – for outding messages (domoticz send message when change any device)
- domoticz/in – for incoming messages (incoming message can influence devices on domoticz)

Send data from MQTT do influxDB (by python script)

Commonly for this purpose is used program telegram from www.influxdata.com. Bat this program have problem to convert some data from domoticz mqtt messages. Therefore we write own python script who read messages from MQTT change format and send into influxDB.

If you need change connection informations, than open and edit this script. This script must be run all of time. If you need run automatical after start, put this execute into */etc/rc.local* .

Donwload this script on ….. .

Use GRAFANA to draw graphs

Description:

Grafana is open source web based program for draw nice graphs from various data source.
In this document is described only basic control how-to create you first graph.

Install:

On ubuntu or debian (only 64-bit):

```
wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_5.1.3_amd64.deb
sudo dpkg -i grafana_5.1.3_amd64.deb
```

Oficial grafana not support for 32-bit and arm platform. For raspberry PI and other amr platform use instalation from: https://github.com/fg2it/grafana-on-raspberry

Start Grafana:

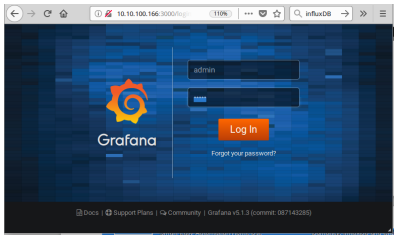
```
/etc/init.d/grafana start
```

Stanadard tcp port for Grafana is 3000. Open your web browser into addres
http://your_garfana_server:3000

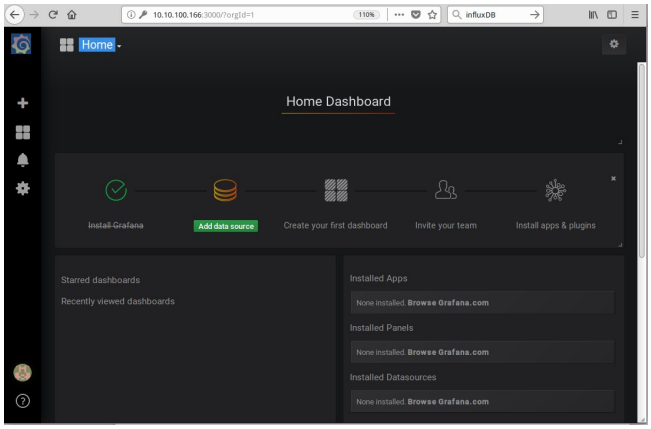
defaut login: admin
defult password: admin

Example howto create first graph

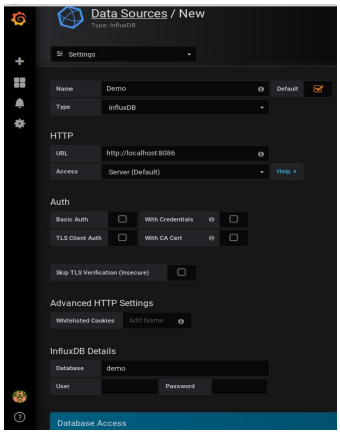
1. login



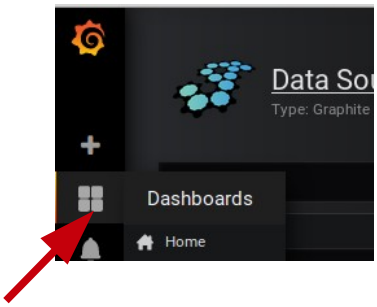
2. Add data source



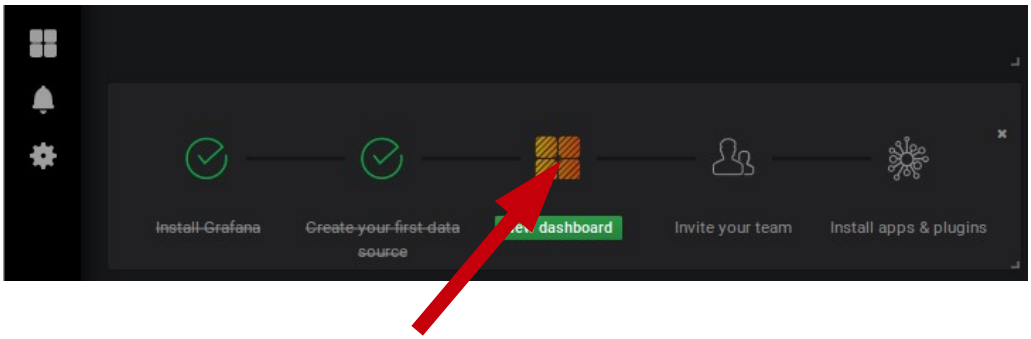
3. select influxDB and fill connection informations



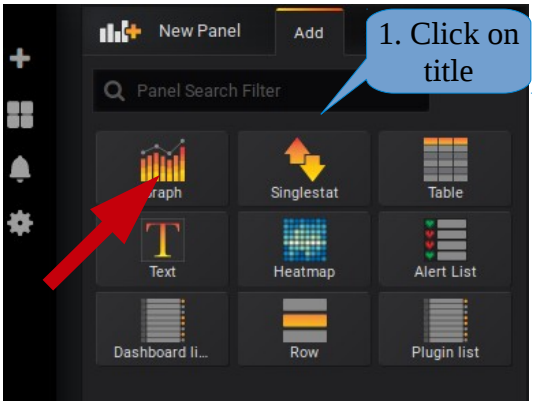
4. After sucessfully save and test go back to main menu



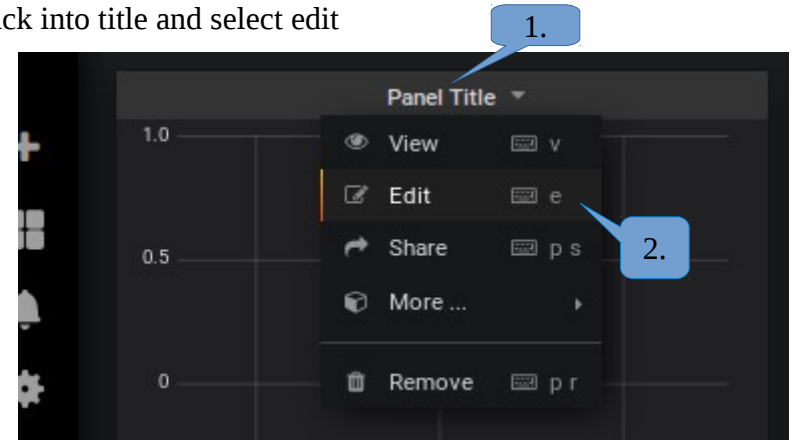
5. Select new dashboard



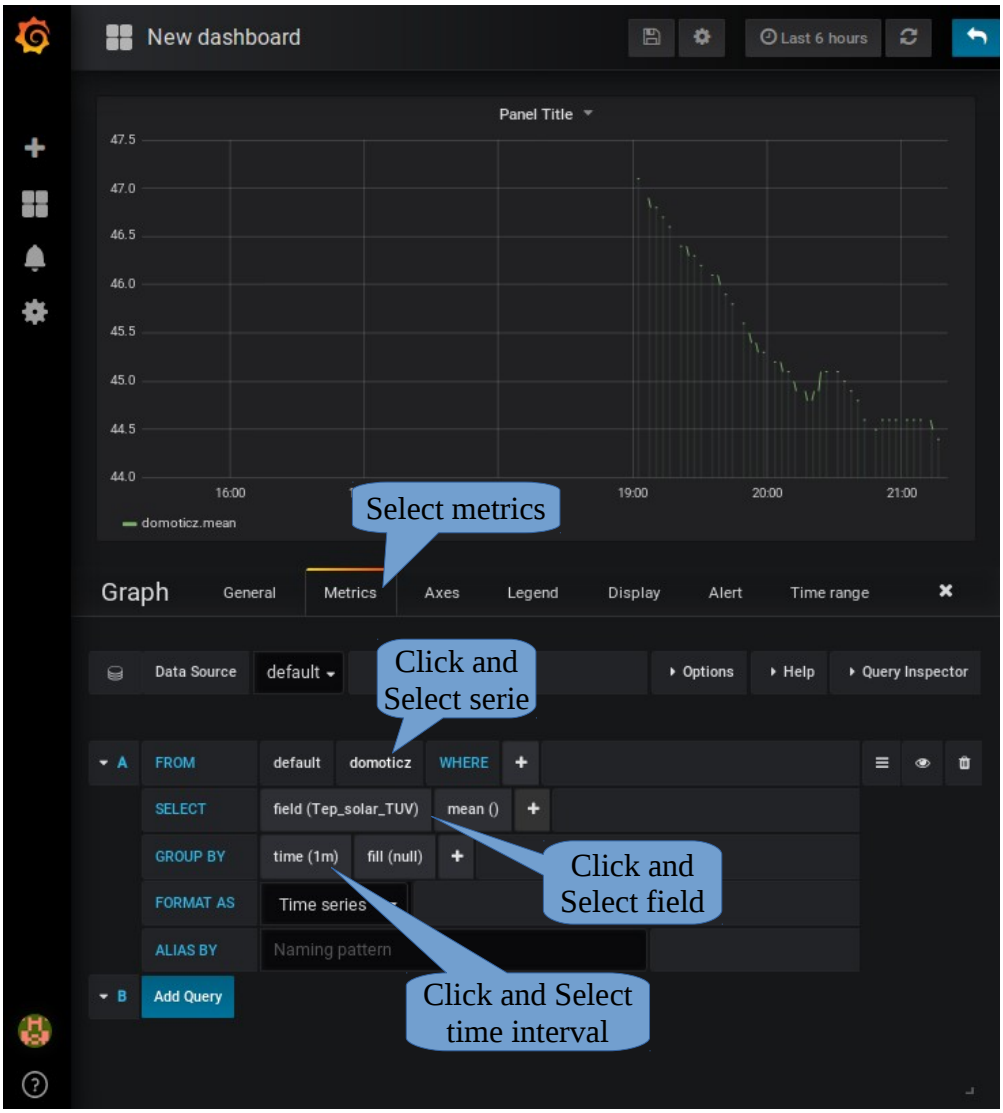
6. Into new dashboard select add graph



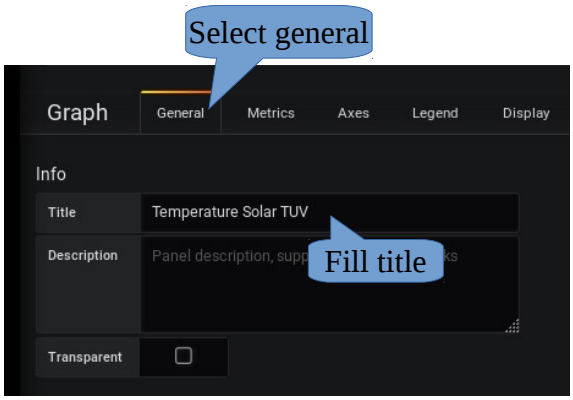
7. In new graph click into title and select edit



8. Edit selection query to data from influxDB source, siply to click on items



9. Fill title of graph



10. Otinaly change display otions and many other graph properties

11. Save graph setting

Now you have you created our first graph. You can add other graphs, tables,.. define users, team, more dashboard.
More documentions: http://docs.grafana.org/

More information and sources:

- mosquitto (MQTT server): https://mosquitto.org/
- domoticz: https://domoticz.com/
- web seahu: http://www.seahu.cz

- Grafana and InfluxDBquickstart on Ubuntu
http://www.andremiller.net/content/grafana-and-influxdb-quickstart-on-ubuntu

-How to Install InfluxDB on Ubuntu 14.04
https://hostpresto.com/community/tutorials/how-to-install-influxdb-on-ubuntu-14-04/

Write by:

Ing. Ondřej Lyčka 5-2018

Document version: 1.01 (drawt)