DATALOGGING MISSION
CITY OF LORIENT, BRITTANY, FRANCE
-
HOWTO MONITOR AND DISPLAY A TEMPERATURE WITH A RASPBERRY PI

Content

1. Framework of the document

The city of Lorient is involved in the Interreg Europe EMPOWER program, which aims to develop energy monitoring systems and associated financial mechanisms to reduce the energy consumption of mid-sized buildings.

As part of this energy monitoring policy, a mission to develop datalogging systems was initiated. This mission is carried out and developed by its environmental department.

The documentation presented here is used as a concept test. It allows to discover and explore a simple datalogging solution in order to verify its technical feasibility before carrying out an on-site deployment.

We will apply our case study to temperature monitoring: from installing an operating system on a Raspberry Pi, to visualizing the data on a monitoring software.

A strong emphasis has been placed on the use of open source software to ensure accessibility and reproducibility of the solution.

This procedure will be carried out in three steps:

- We will configure a Raspberry Pi and its operating system Raspbian.
- We will configure a DS18B20 temperature sensor on our Raspberry Pi to read, display, and record temperature measurements.
- We will install and configure monitoring tools, namely Zabbix and Grafana, to graphically monitor these temperatures.

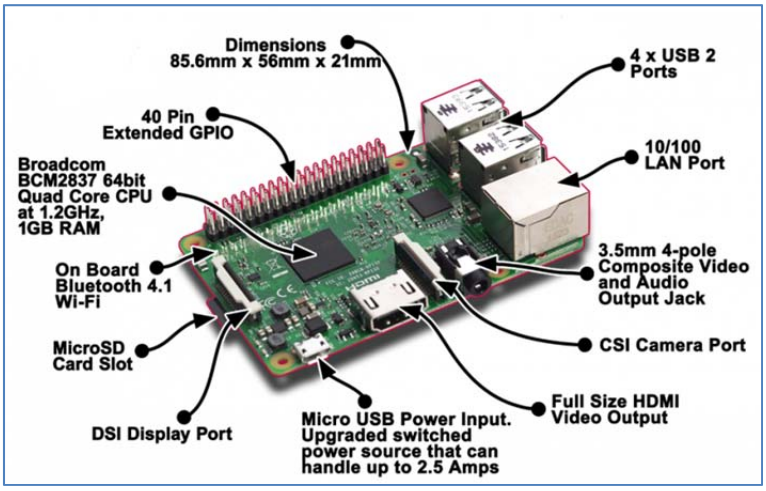
2. Install and configure a Raspberry Pi

This first part describes the installation and configuration steps of an operating system and additional administration tools required for a Raspberry Pi.

2.1. Overview of the Raspberry Pi

The Raspberry Pi is a tiny and affordable computer that you can use to learn programming through fun, practical projects. It is an ideal tool for the development of Do It Yourself projects in the areas of home-automation, multi-media, etc.

In this procedure, we will use a Raspberry Pi 3 Model B v1.2 de 2015 with a Quad-core ARM Cortex-A53 1.2 GHz and 1024Mo de RAM.



We will use the free and open source Raspbian operating system based on Debian GNU/Linux and optimized to run on a Raspberry Pi.

Raspbian

More information on: https://www.raspberrypi.org/.

To perform this procedure, you will need:

- A Micro SD card and possibly a Micro SD card to SD card adapter plug.
- A Micro USB power supply (2.1 A).
- An Ethernet cable.

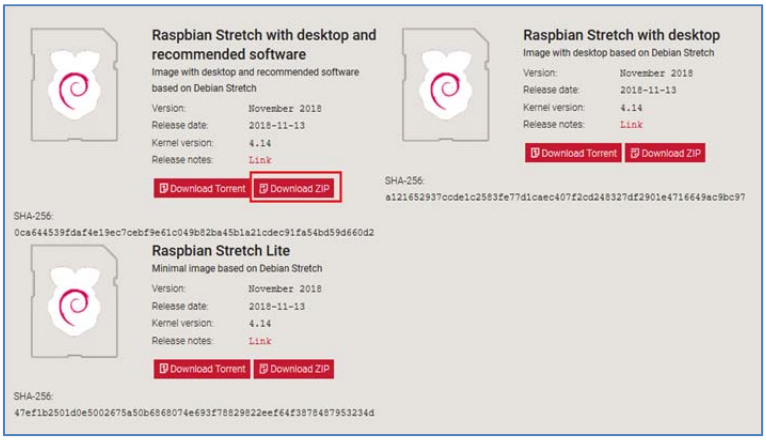
To use the Raspbian desktop (graphical user interface):

- An HDMI cable and a monitor.
- A keyboard and a mouse.

2.2. Installing Raspbian

2.2.1. Downloading Raspbian

Download an official version of Raspbian at https://www.raspberrypi.org/downloads/ and select the image "Raspbian Stretch with desktop and recommended software" in.zip format.



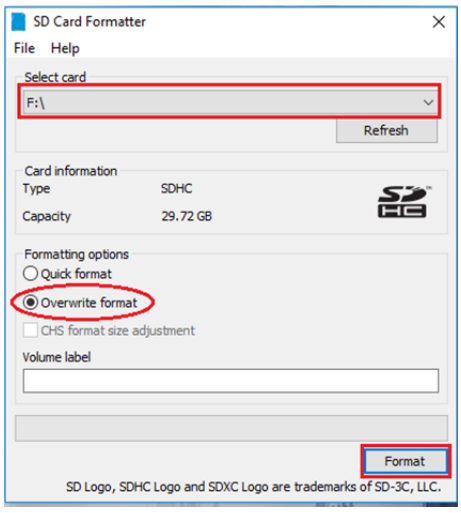
2.2.2. Format your SD Card

For the next steps, you will need a computer with an SD card reader to write the Raspbian image to your Micro SD card.

It is a good idea to format your SD card before copying the installation files. Even, if it is brand new. The use of the formatting software developed by the SD Association is also recommended.

The software is available for Windows at: https://www.sdcard.org/downloads/formatter_4/eula_windows/index.html/

Select the formatting options on « SD Card Formatter »:

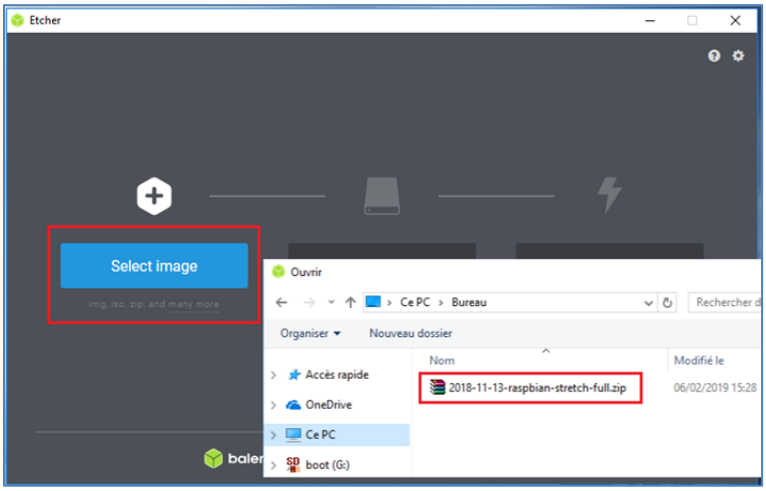


2.2.3. Burning Raspbian on the SD card

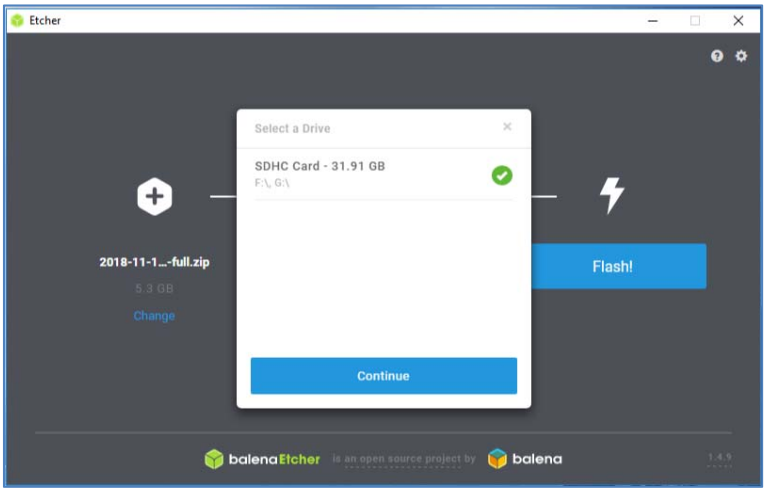
In order to prepare the Raspbian SD card, it is recommended to use the Etcher software available at: https://www.balena.io/etcher/.

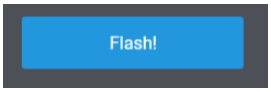
Etcher is a free software developed with Electron (framework), for burning images (ZIP, img, iso) on different USB key media, SD Card, for GNU/Linux, Windows, MacOS. The application is portable and has a graphical interface.
Do this:

- Download and install Etcher.
- Insert your Micro SD card into an SD card reader (may require a micro SD card to SD card adapter.
- Open Etcher and select from your hard drive the Raspbian.img or Raspbian.zip that you want to write on your SD card:

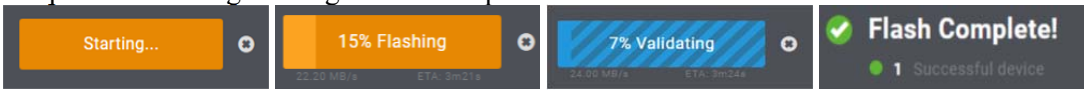


- Select your SD card :



- Review your selected options and click on 

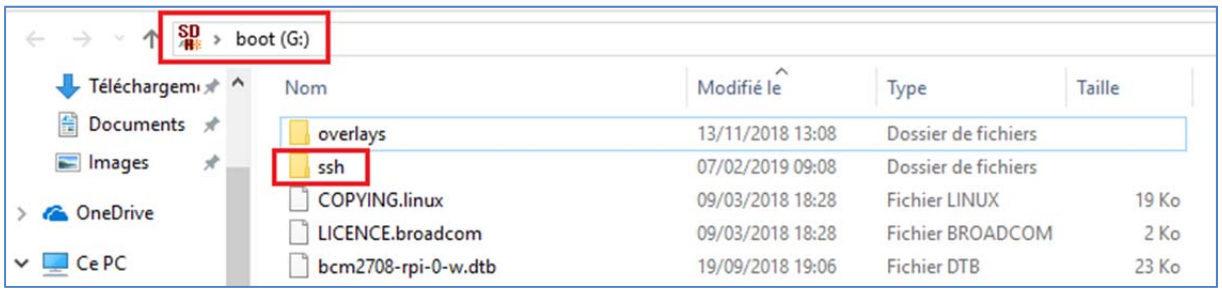
to start writing Raspbian on your SD card.
- The procedure will go through several steps:



2.2.4. Enabling SSH access

Administering your Raspberry Pi remotely via SSH may be necessary if you want to use it as a "server". SSH or "Secure Shell" is a secure communication protocol.

The SSH server is disabled by default on Raspbian. To activate it, simply create an empty folder named "ssh" in the "boot" partition of your SD card. At the first start the file will be automatically detected by the system, the SSH server will be activated and the file will be deleted.



We will see further in this procedure how to connect to a Raspberry Pi from a PC using Windows in SSH with the PuTTY utility program.

2.3. Taking control of your Raspberry Pi

2.3.1. First boot of your Raspberry Pi

At first boot, it is best to have a monitor with a HDMI connection, a keyboard and a mouse to run configurations using the graphical user interface.

A Raspberry Pi 3 Model B v1.2 powered up:

Caution
There is no “start” button on the Raspberry Pi. As soon as it is plugged in, Raspbian is executed.
In addition, it is strongly discouraged, in order not to corrupt the SD card and your data, to disconnect the power supply when the Raspberry Pi is in use.

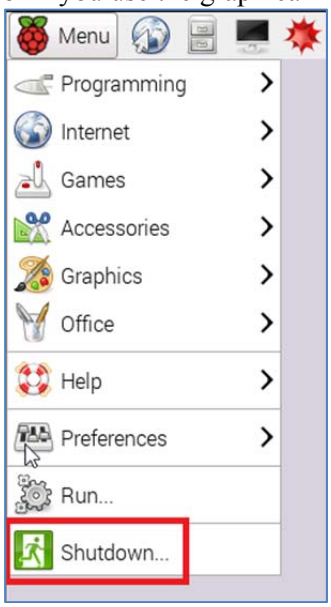
To switch it off correctly using the terminal, run the command:

```
sudo shutdown -h now
```

or

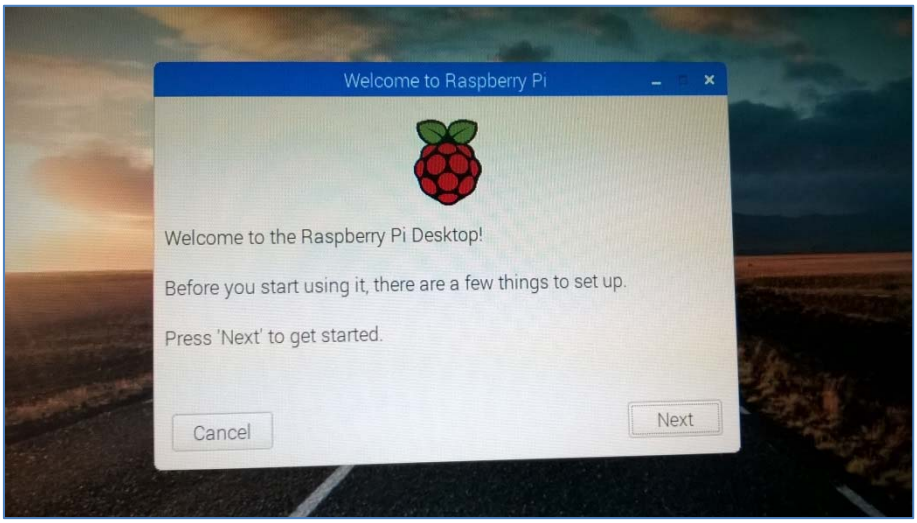
```
sudo halt
```

or if you use the graphical interface:



You can then disconnect the Raspberry Pi.

At first boot, using the graphical user interface, you should see this screen:

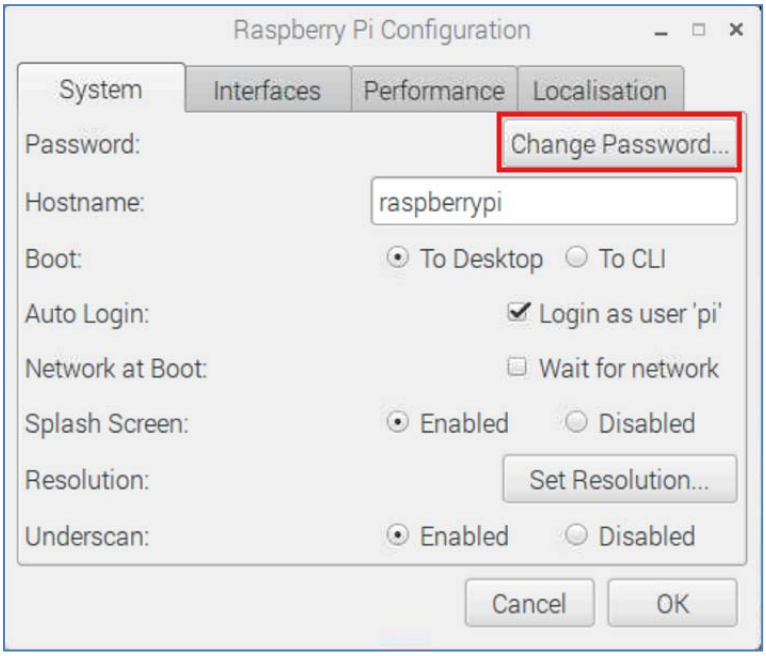


At first boot, the default login is "pi", and the password is "raspberry". You should change it as soon as possible, leaving it, is a security vulnerability.

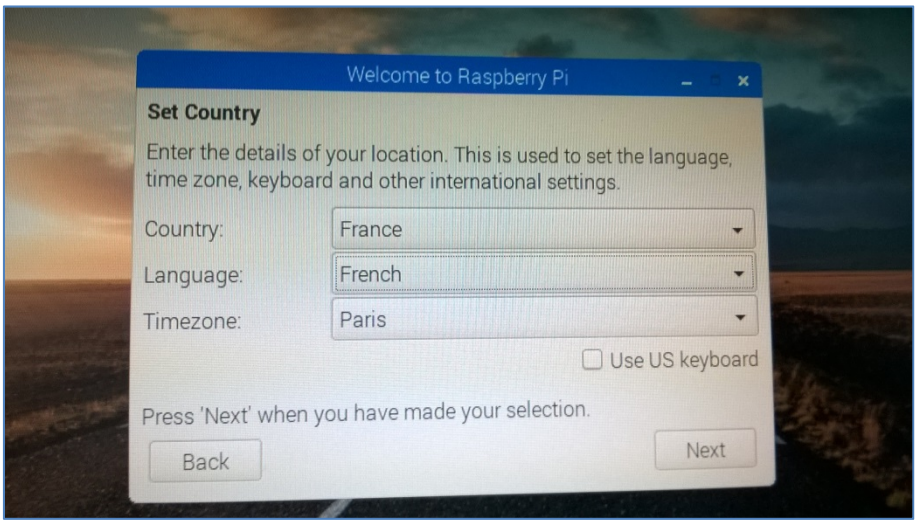
The "sudo" command allows you to launch a command as an administrator or otherwise named "root".

If you were not prompted to change it during first boot (see capture below) go to "Menu > Preferences > Raspberry Pi Configuration > System > Change User Password", and confirm. A window informs you that you will need to enter a new password for the user "pi":

Alternatively, below, the password change window in the "Configuration" menu:



Next choose your location and language. Then, follow the rest of the instructions:



2.3.2. Updating your Raspberry Pi

Your system is now configured. However, you should update it to minimize security breaches and bugs.

There are three elements to regularly update on a Raspberry Pi:
- Packages,
- The Distribution (Raspbian),
- The Firmware.

Updating these components requires an internet connection.

- **Updating packages**

```
sudo apt-get update
sudo apt-get upgrade
```

- **Upgrading Raspbian**

```
sudo apt-get dist-upgrade
```

- **Updating the Firmware**

Install the “rpi-update” utility software:

```
sudo apt-get update
sudo apt-get install rpi-update
```

Run the utility to perform the update:

```
sudo rpi-update
```

Reboot the system:

```
sudo reboot
```

- **Miscellaneous: Git repository manager and I2C**

Install the git command allows you to retrieve or synchronize source code on GitHub. It may be useful to you in future proceedings:

```
sudo apt-get install git
```

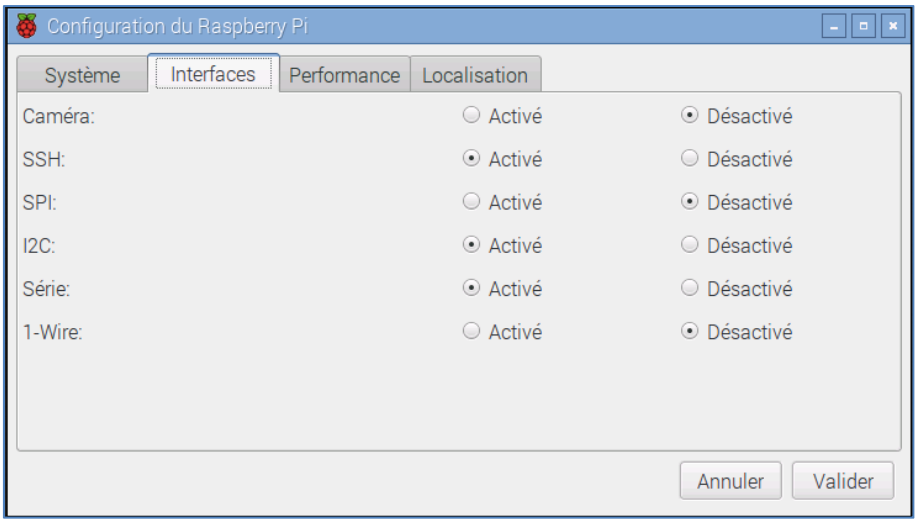
Install tools to manage the I2C bus. There are many devices that use this I2C bus for home automation and electronics applications (e. g. temperature, humidity module, etc.):

```
apt-get install i2c-tools
```

2.3.3. How to connect using SSH with PuTTY

First, check that the SSH connection is enabled.

To do this, you can open the Raspberry Pi Configuration utility in "Menu > Preferences > Raspberry Pi Configuration > Interfaces (tab)" and check the line "SSH: On".



Or, check its activation using the terminal:

```
ssh –V
```

To activate SSH from the terminal, run:

```
sudo systemctl start ssh
```

On Windows, the use of a third-party program, such as PuTTY, available at https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html/, is required to initiate a SSH connection and execute command prompt.

It is necessary to know the IP address of your Raspberry Pi. You can obtain by running:

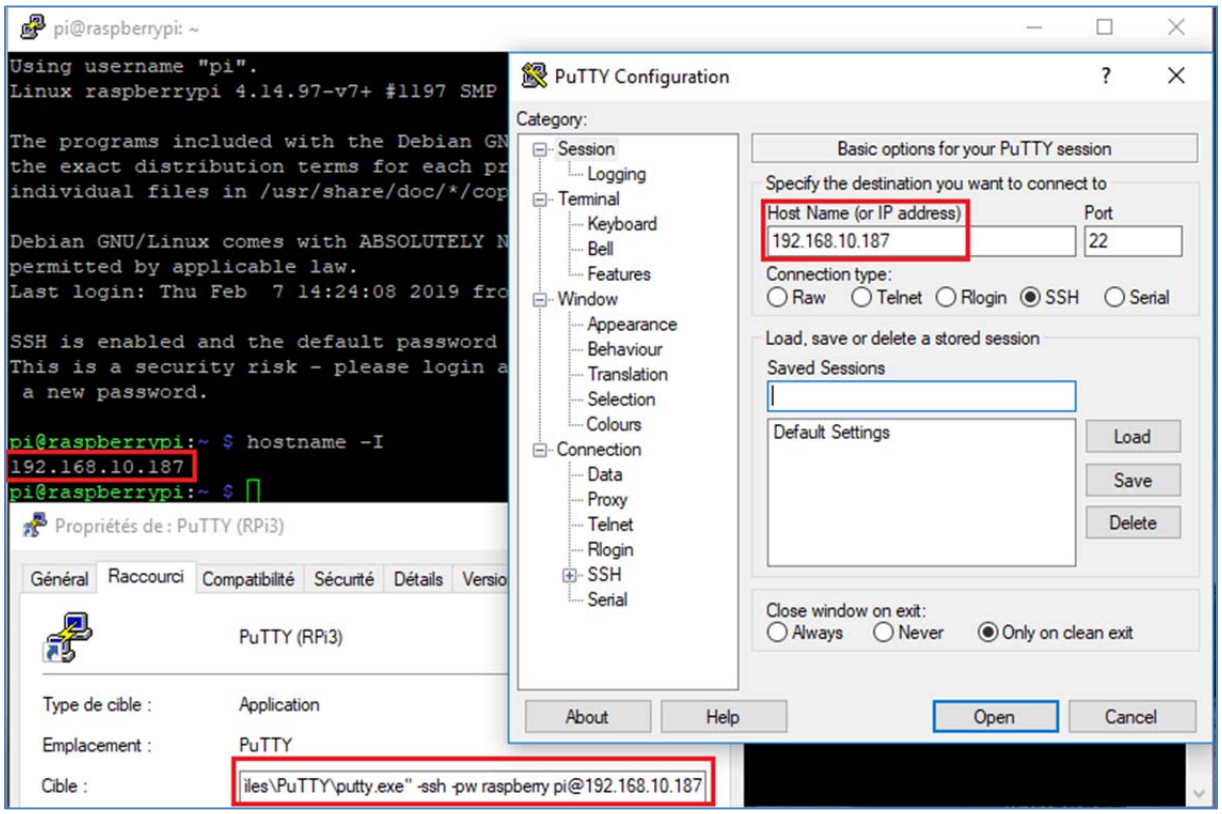
```
ifconfig
```

or

```
hostname -I
```

On the PuTTY GUI, simply launch the connection using the "Open" button, then enter the user's name and password (by default on Raspbian the user is "pi" and his password is "raspberry").

If you have not changed your password at this point, do so, as your Raspberry Pi will be visible on your network and remotely accessible if another user knows the default password for a Raspberry Pi.



You can also create a PuTTY shortcut by changing the "target" in the shortcut property (see screenshot above).

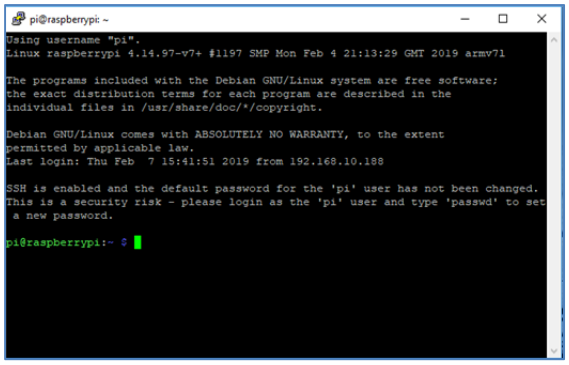
Follow this template to create a shortcut:

```
putty.exe" -ssh -pw user_password user_name@ip_adress
```

In our case:

```
putty.exe" -ssh -pw raspberry pi@192.168.10.187
```

When launching PuTTY from your PC under Windows, you should get a similar window:



It is now possible to remotely install packages or modify files using your PC to execute command prompt.

2.3.4. How to connect using FTP with FileZilla

It can be useful to remotely be able to drop or retrieve Python or PHP scripts and modify them in a text editor using FileZilla and the FTP Protocol.

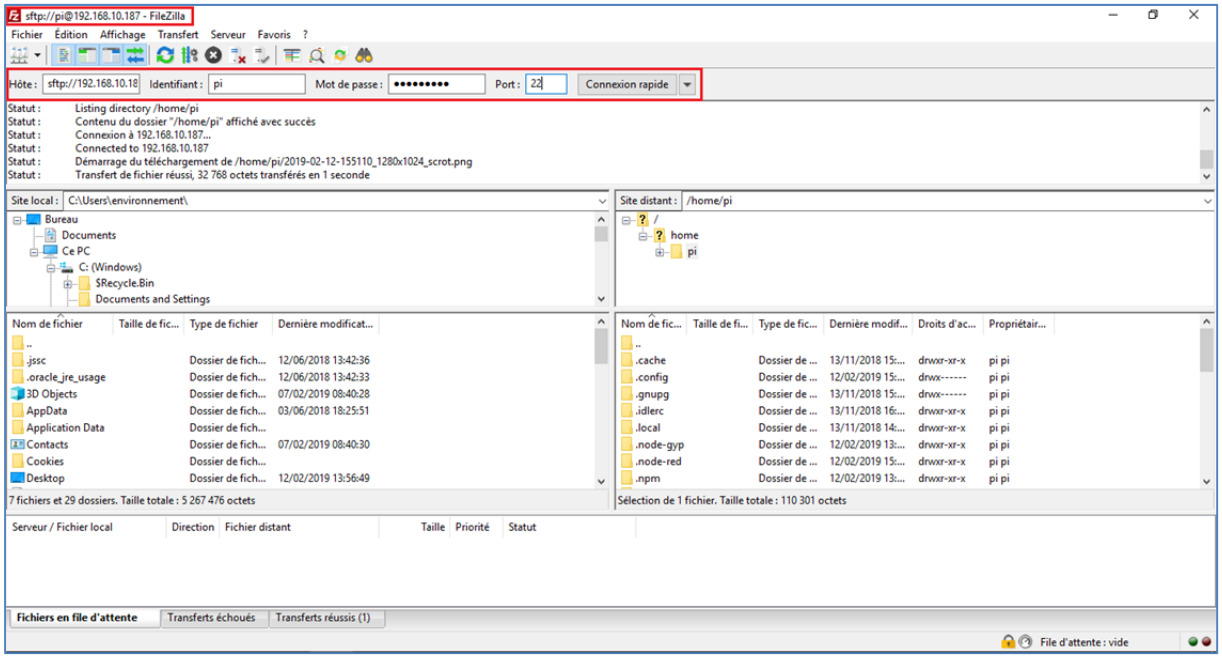
FileZilla Client is a FTP, FTPS and SFTP client, developed under the GNU General Public License. To download the client version of the program, go to: https://filezilla-project.org/.

Once installed and started, configure the connection like this:

- Host: the IP address of the Raspberry.
- Login: by default "pi".
- Password: the user's password "pi". By default "raspberry".
- Port: 22 (the SSH port).
- Press the "quick connect" button.

Note that your SSH connection must be enabled on Raspbian.

Once connected, you will find on the left the hierarchy tree of your PC and on the right the one of the Raspberry Pi.



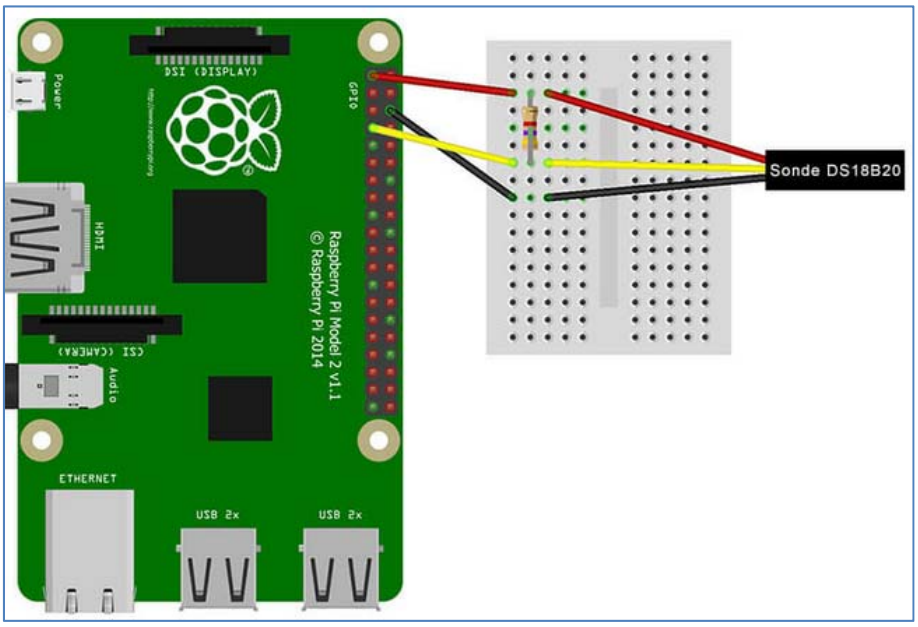
3. How to read a temperature with the DS18B20 sensor

This second part is aimed at installing and configuring a 1-Wire DS18B20 sensor to read and log a temperature.

3.1. Overview of the DS18B20 sensor

The DS18B20 is a digital temperature sensor communicating via a 1-Wire bus with a measuring range from -55°C to +125°C accuracy ±0.5°C between -10°C and +85°C.

The sensor is already pre-wired with resistance of 4.7 kΩ (yellow, purple, red) placed on the red and yellow wires. The sensor only has to be connected directly to the GPIO port of the Raspberry Pi according to the following diagram:



Below, a picture of the DS18B20 probe connected to a Raspberry Pi:

3.2. Configuring your Raspberry Pi

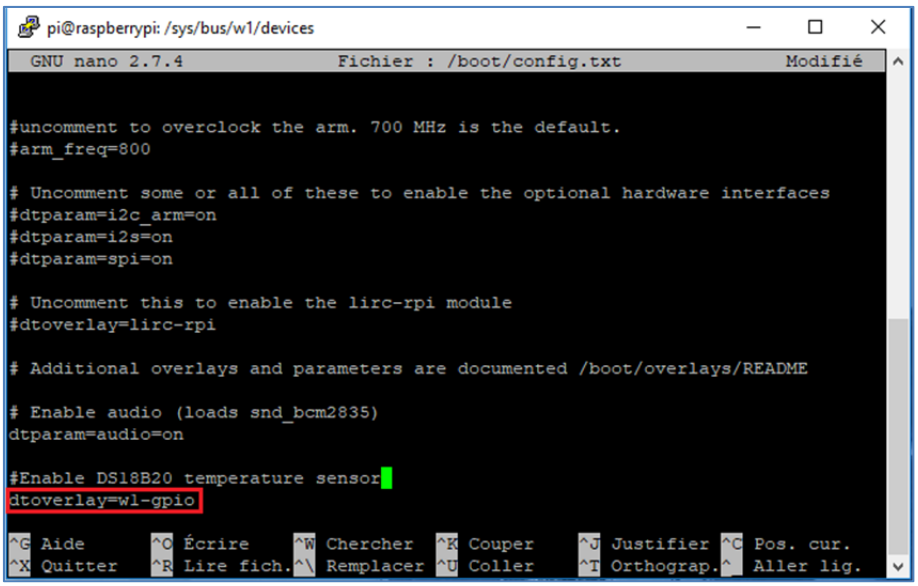
The sensor is not automatically recognized by the system. Before it can be used, it is necessary to configure the system to be able to communicate with it.

From your terminal, edit the file "/boot/config.txt":

```
sudo nano /boot/config.txt
```

Add the following lines at the end of the file:

```
dtoverlay=w1-gpio
```



Save the file with "CTRL+W", and exit with "CTRL+X".

Reboot the system:

```
sudo reboot
```

To load the kernel modules required to use the sensor, run the following commands:

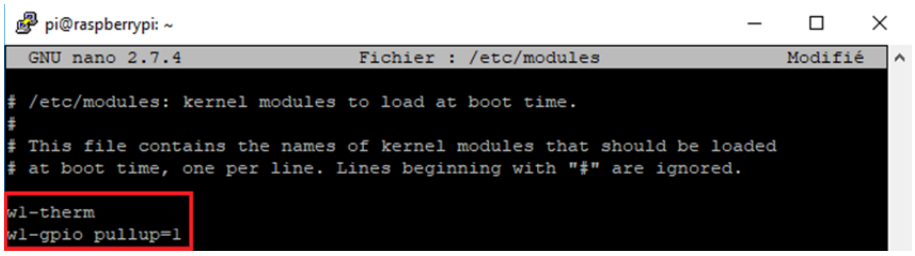
```
sudo modprobe w1-therm
sudo modprobe w1-gpio
```

In order for these modules to be automatically loaded at system boot, you must modify the following file "/etc/modules":

```
sudo nano /etc/modules
```

Add the two following lines at the end of the file:

```
w1-therm
w1-gpio pullup=1
```



3.3. First readings

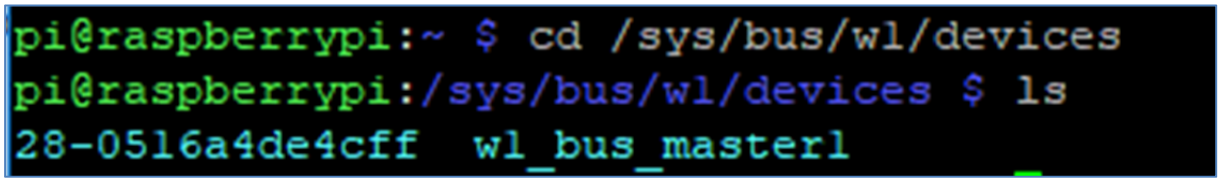
To read a temperature in your terminal, simply go to the folder "/sys/bus/w1/devices/" and then look for the file starting with "28-*".

Move to the directory « /sys/bus/w1/devices »:

```
cd /sys/bus/w1/devices
```

To display the list of devices in this directory, run:

```
ls
```



The serial number of your sensor should be displayed:

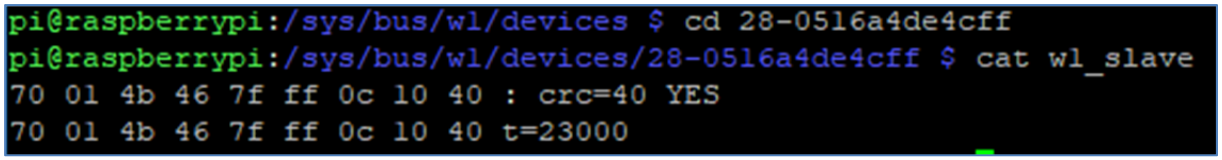
```
28-0516a4de4cff  w1_bus_master1
```

To query this device, we must go to its directory. Change the X's by your own serial number:

```
cd 28-XXXXXXXXXXXX
```

Then, run the command "cat" on the file "w1_slave". This command will simply read the contents of the file to you:

```
cat w1_slave
```



"YES" means that your Raspberry Pi can communicate with your sensor.
The temperature is expressed in Celsius degrees, multiplied by 1000. Here, the temperature is 23,000°C. It is represented as "t=23000" at the end of the file.

3.4. Display a temperature in a loop

Go back to the "root" directory:

```
cd
```

Create a new directory named "tempLog":

```
mkdir tempLog
```

Move into the new directory and create a new Python file:

```
cd tempLog
sudo nano getTemp.py
```

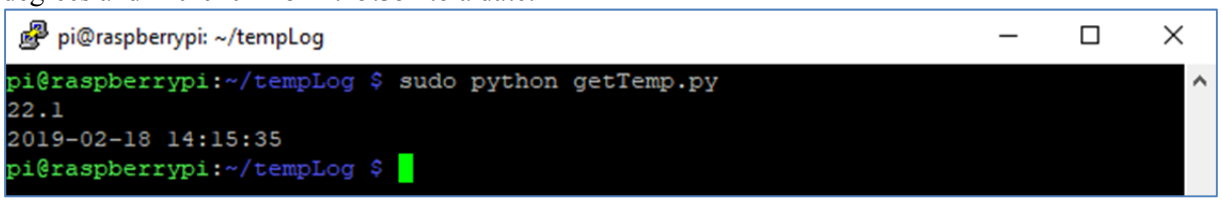
This program displays temperature readings on an SSH terminal. Remember to use the serial number of your sensor.

```
#SCRIPT TO RECOVER DATAS FROM THE DS18B20 TEMPERATURE PROBE
#DISPLAY THE TEMPERATURE IN CELSIUS DEGREE

# -*- coding: utf-8 -*-

#IMPORT ALL OF THE PYTHON LIBRARIES WE NEED
import os
import time
import datetime
import glob
from time import strftime

#LOAD THE GPIO AND THERM KERNELS AGAIN USING MODPROBE
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
#GET OUR PROBE NUMBER STARTING WITH 28-*
temp_sensor = '/sys/bus/w1/devices/28-0516a4de4cff/w1_slave'

def tempRead():
            #READ IN THE w1_slave FILE USING open 'r'
        t = open(temp_sensor, 'r')
        lines = t.readlines()
        t.close()

        temp_output = lines[1].find('t=')
        if temp_output != -1:
                        #IF RETURNED VALUE IS -1, IT DID NOT FIND IT
                        #WITHOUT THE +2 YOU GET 'T=XXXXX'
                temp_string = lines[1].strip()[temp_output+2:]
                temp_c = float(temp_string)/1000.0
        return round(temp_c,1)

while True:
    temp = tempRead()
    print temp
    datetimeWrite       =       (time.strftime("%Y-%m-%d       ")       +
time.strftime("%H:%M:%S"))
    print datetimeWrite
      #CAN REMOVE BREAK AND INSERT TIME.SLEEP(60)
      #TO OUTPUT THE TEMP AND CURRENT DATE AND TIME EVERY MINUTE
    break
```

In the "tempLog" directory, run:

```
sudo python getTemp.py
```

You should see numbers displayed in your terminal "22.1" corresponding to a temperature in Celsius degrees and "2019-02-18 14:15:35" to a date:



3.5. Log a temperature in a database

3.5.1. Installing a LAMP server

To log our temperature values, we will need a database using MySQL. To access the database online, we will need a web server and a programming language to run it, Apache and PHP respectively. Therefore, we will use a LAMP architecture.

LAMP is an acronym for Linux, Apache, MySQL, and PHP. It is a software stack that includes the operating system, an HTTP server, a database management system and an interpreted programming language, and allows you to set up a web server.

- **Installing Apache**

Check that your Raspbian is up to date:

```
sudo apt update
sudo apt upgrade
```

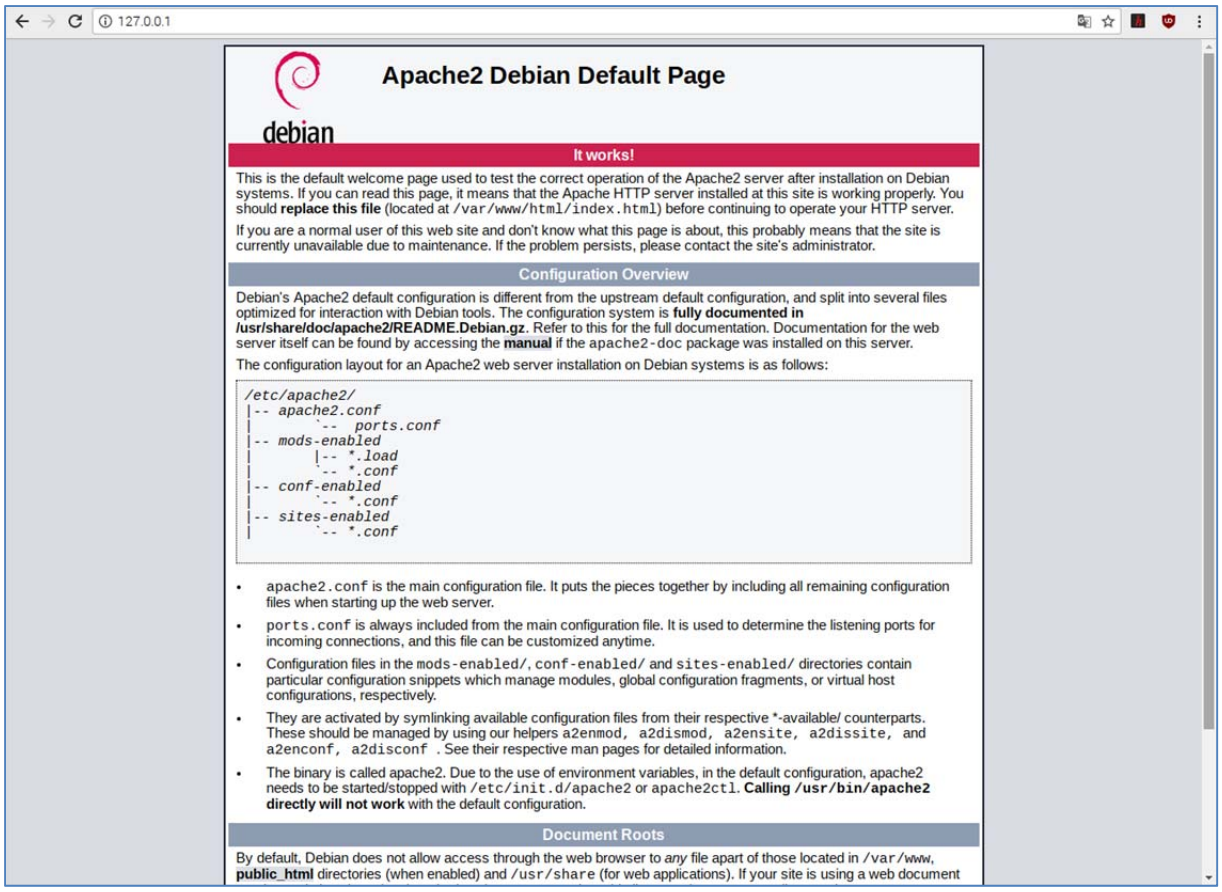
To install Apache, run the following command. When prompted, type "y":

```
sudo apt install apache2
```

Give rights to the apache folder that will allow you to administer the sites. To do this, run the following commands:

```
sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
```

Once the installation is complete, make sure that Apache is working properly by going to "http://127.0.0.1" from your Raspberry Pi browser. You should get a page with the following message: "It works!".



Alternatively, you can run the following command in your terminal:

```
wget -O verif_apache.html http://127.0.0.1
```

This command will save the HTML code of the page in the "verif_apache.html" file in the current directory. Then, read the file:

```
cat ./verif_apache.html
```

If you see "It works!" displayed in the code, it means that Apache is working.

- **Installing PHP**

Update PHP and its packages by running:

```
apt-get install php*cli
sudo apt install php-curl php-gd php-intl php-json php-mbstring php-xml php-zip
```

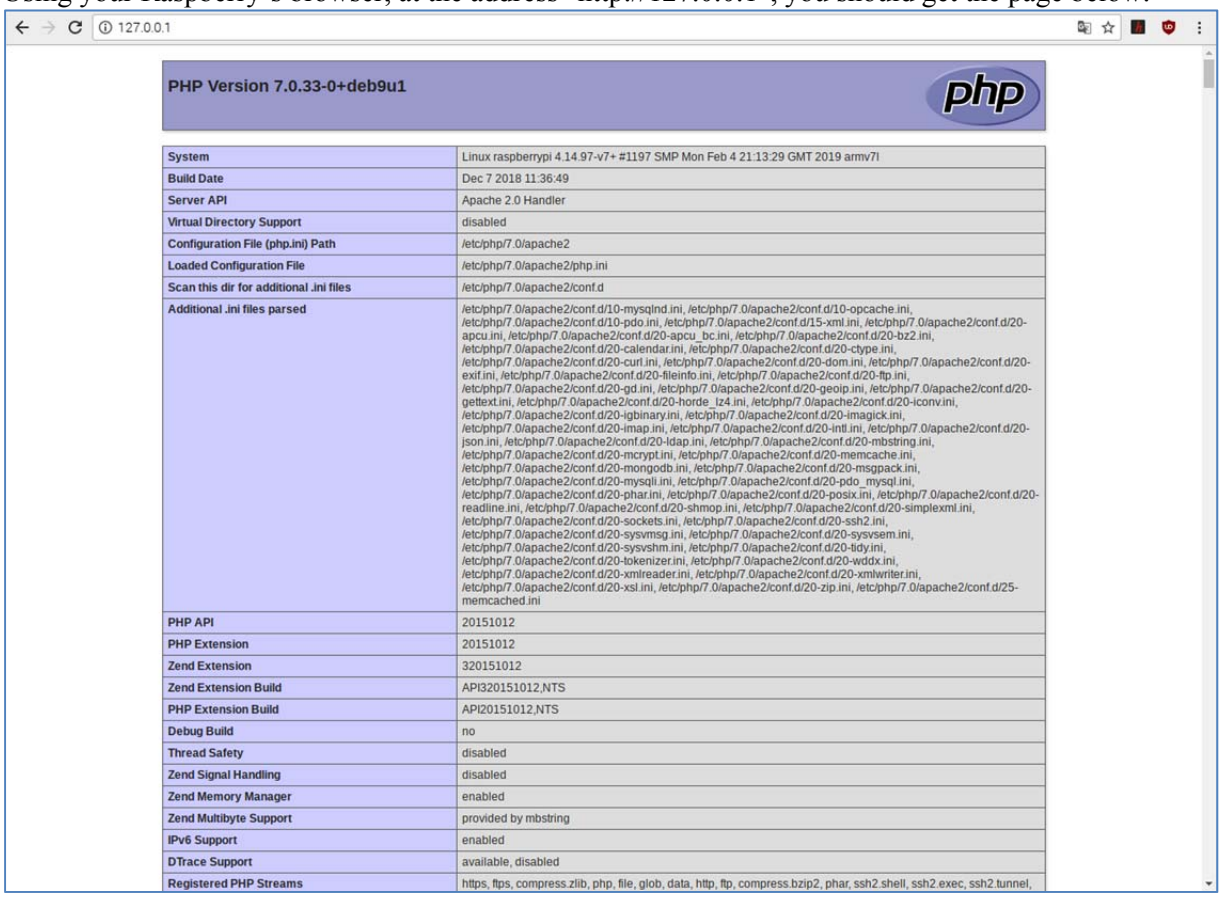
To check if PHP is working, delete the “index.html” in the directory "/var/www/html":

```
sudo rm /var/www/html/index.html
```

Then create a new "index.html" in the same directory:

```
echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```

Using your Raspberry's browser, at the address "http://127.0.0.1", you should get the page below:



PHP Version 7.0.33-0+deb9u1

System	Linux raspberrypi 4.14.97-v7+ #1197 SMP Mon Feb 4 21:13:29 GMT 2019 armv7l
Build Date	Dec 7 2018 11:36:49
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-apcu.ini, /etc/php/7.0/apache2/conf.d/20-apcu_bc.ini, /etc/php/7.0/apache2/conf.d/20-bz2.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-curl.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gd.ini, /etc/php/7.0/apache2/conf.d/20-geoip.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-horde_lz4.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-igbinary.ini, /etc/php/7.0/apache2/conf.d/20-imagick.ini, /etc/php/7.0/apache2/conf.d/20-imap.ini, /etc/php/7.0/apache2/conf.d/20-intl.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-ldap.ini, /etc/php/7.0/apache2/conf.d/20-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-memcache.ini, /etc/php/7.0/apache2/conf.d/20-mongodb.ini, /etc/php/7.0/apache2/conf.d/20-msgpack.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-ssh2.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tidy.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini, /etc/php/7.0/apache2/conf.d/20-zip.ini, /etc/php/7.0/apache2/conf.d/25-memcached.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, ssh2.shell, ssh2.exec, ssh2.tunnel,

- **Installing MySQL**

In order to log our temperature measurements we will have to set up a DBMS (Database Management System), namely MySQL.

Install MySQL with the following command. Enter "y" when prompted:

```
sudo apt install mysql-server php-mysql
```

A prompt appears asking you to enter a password for the MySQL database. Memorize it.

At some point we will edit our MySQL database from a Python script, so download the corresponding Python library:

```
sudo apt-get install python-mysqldb
```

- **Creating a databse using MySQL**

Launch MySQL:

```
sudo mysql –u –p
```

This connects us to MySQL as a "root" user (-u) and asks us for a password (-p). Enter the password you previously created for MySQL.

We will now delete the “root” user (DROP USER) and create a new "root" user (CREATE USER), because the default one is only usable by the system administrator account, and is therefore not accessible to PHP scripts on the server. We then grant it all the privileges (GRANT ALL PRIVILEGES) on all databases (*.*).

```
DROP USER 'root'@'localhost';
CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost';
```

The next time you connect, use the command:

```
mysql --user=root --password=yourpassword
```

We will now create a new database:

```
CREATE DATABASE temp_database;
```

Check if the database was successfully created:

```
SHOW DATABASES;
```

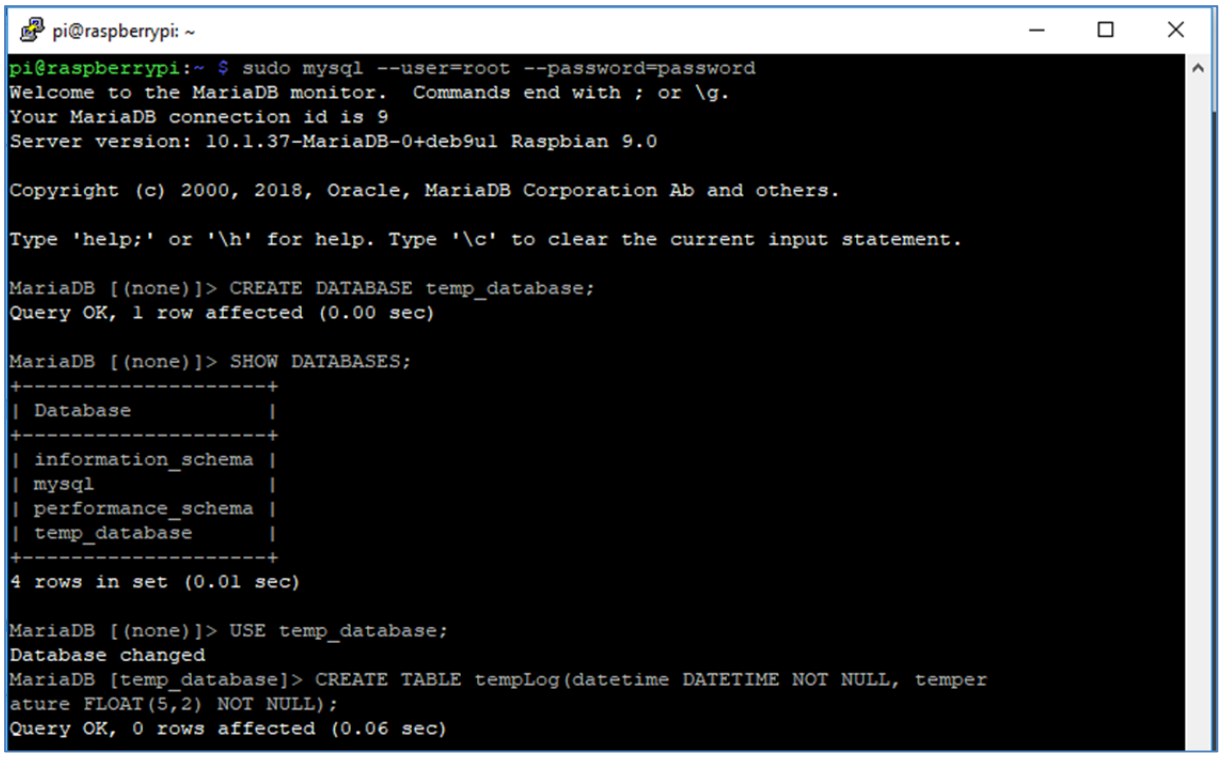
A list of databases currently hosted by MySQL is displayed.

We want to create a new table in the "temp_database" database. To do this, we must first tell MySQL that we want to use the “temp_database” database:

```
USE temp_database;
```

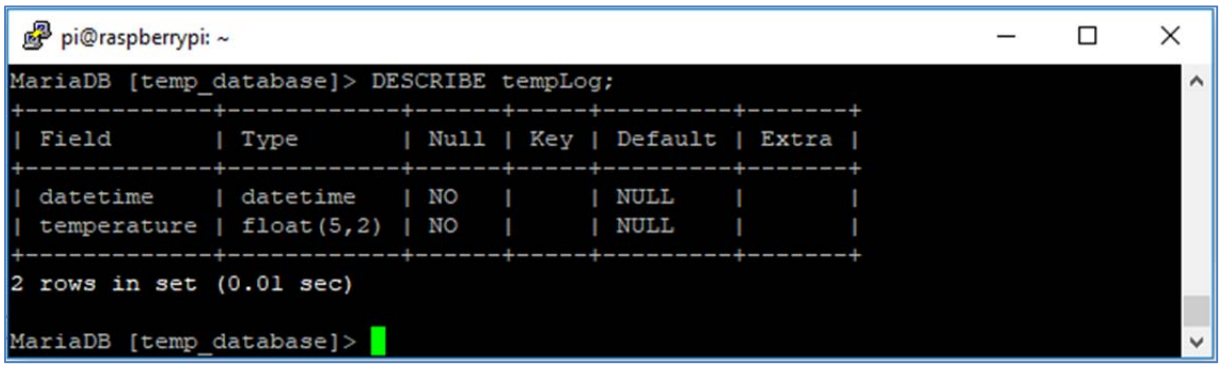
We want to create a table in MySQL with two non-null fields: "datetime" (DATETIME type) and "temperature" (FLOAT type):

```
CREATE TABLE tempLog(datetime DATETIME NOT NULL, temperature FLOAT(5,2) NOT NULL);
```



Check if the table was successfully created:

```
DESCRIBE tempLog;
```



Exit MySQL with "CTRL+Z".

- **Installing phpMyAdmin**

PhpMyAdmin is a web-based management application for MySQL DBMS that allows you to easily manage the content of your databases, without having to write your own SQL queries.

To install it, run:

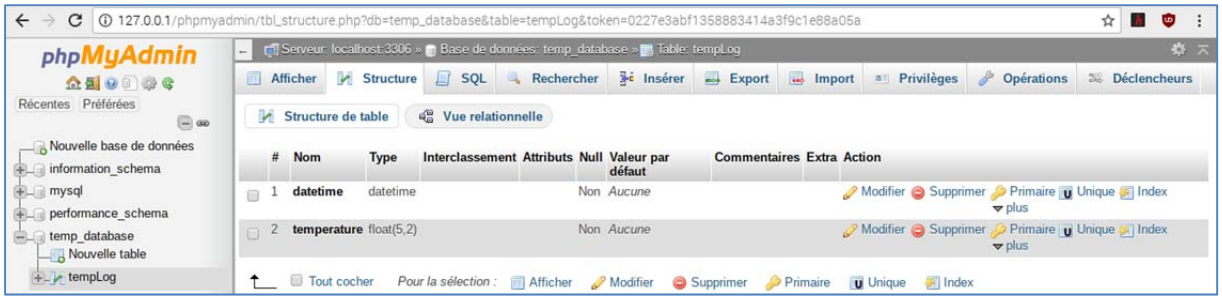
```
sudo apt install phpmyadmin
```

During the installation of phpMyAdmin, you will be asked several questions about its settings. As we have already configured the database, choose “no” when asked about using "dbconfig-common". Choose to use phpMyAdmin for an Apache server. For the "root" password, it is the one you used for MySQL.

You can now access phpMyAdmin and your databases from your browser by going to "127.0.0.0.1/phpmyadmin/":

If an error is displayed, it may be due to the fact that phpMyAdmin is installed in the wrong folder. In this case, run the following command:

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```



3.5.2. Communicating with MySQL

Let's rework our previous Python script in order to write the date and temperature in our MySQL database "temp_database". Go to the "tempLog" directory and create the script "readTempSQL.py":

```
#SCRIPT TO WRITE DATE AND TEMPERATURE IN MYSQL DB

#!/usr/bin/env python

import os
import time
import datetime
import glob
import MySQLdb
from time import strftime

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
temp_sensor = '/sys/bus/w1/devices/28-0516a4de4cff/w1_slave'

# VARIABLES FOR MYSQL
db   =   MySQLdb.connect(host="localhost",   user="root",passwd="password",
db="temp_database")
cur = db.cursor()

def tempRead():
    t = open(temp_sensor, 'r')
    lines = t.readlines()
    t.close()

    temp_output = lines[1].find('t=')
    if temp_output != -1:
        temp_string = lines[1].strip()[temp_output+2:]
        temp_c = float(temp_string)/1000.0
    return round(temp_c,1)

while True:
    temp = tempRead()
    print temp
    datetimeWrite       =       (time.strftime("%Y-%m-%d       ")       +
time.strftime("%H:%M:%S"))
    print datetimeWrite
      # SQL QUERY TO INSERT DATA INTO THE TABLE
    sql   =   ("""INSERT   INTO   tempLog   (datetime,temperature)   VALUES
(%s,%s)""",(datetimeWrite,temp))
    try:
        print "Writing to database..."
            # RUN THE SQL COMMAND
        cur.execute(*sql)
        # COMMIT YOUR CHANGES INTO THE DATABASE
        db.commit()
        print "Write Complete"

    except:
        # ROLLBACK IF ANY ERROR
        db.rollback()
        print "Failed writing to database"

    cur.close()
    db.close()
    break
```

The variables that are used to communicate with MySQL (password, user, host, etc.) lines 11 and 12 and the SQL query to insert data into the database lines 32 and 33 must be adapted to your situation.

Launch the script several times to write measurements into your database:

```
sudo python readTempSQL.py
```

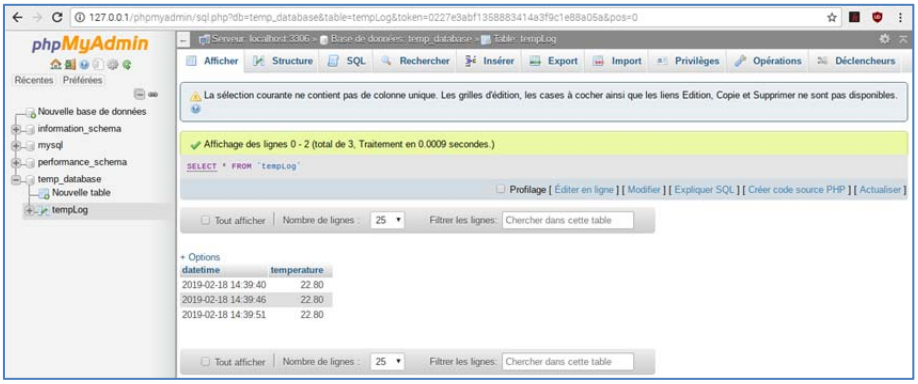
```
pi@raspberrypi:~/tempLog $ sudo python readTempSQL.py
22.8
2019-02-18 14:39:51
Writing to database...
Write Complete
pi@raspberrypi:~/tempLog $
```

Check the "tempLog" table for data:

```
USE temp_database;
SELECT * FROM tempLog;
```

```
pi@raspberrypi: ~/tempLog
pi@raspberrypi:~/tempLog $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 55
Server version: 10.1.37-MariaDB-0+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE temp_database;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [temp_database]> SELECT * FROM tempLog;
+---------------------+-------------+
| datetime            | temperature |
+---------------------+-------------+
| 2019-02-18 14:39:40 |       22.80 |
| 2019-02-18 14:39:46 |       22.80 |
| 2019-02-18 14:39:51 |       22.80 |
+---------------------+-------------+
3 rows in set (0.00 sec)

MariaDB [temp_database]>
```

Alternatively, check it on phpMyAdmin:



3.6. Scheduling an automatic reading

If we want our script to automatically record a temperature reading in the database every T minutes, we will use Crontab, which is a unix task scheduling tool.

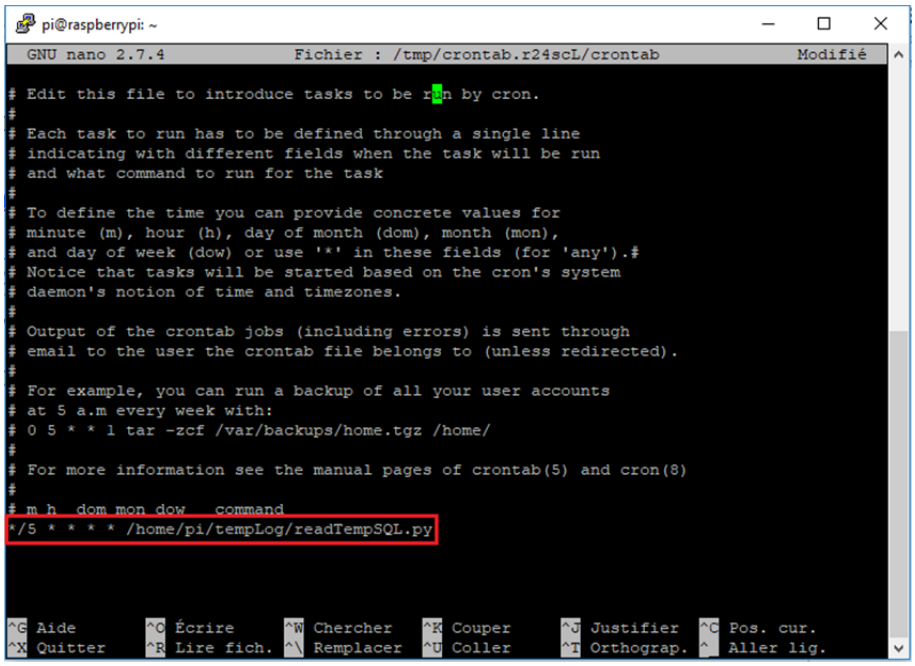
To open Crontab from the "root" directory, run:

```
sudo crontab -e
```

Add the following line at the end of the Crontab file:

```
*/5 * * * * /home/pi/tempLog/readTempSQL.py
```

It simply executes our Python script every 5 minutes. If you have used different file and folder names, change the line accordingly. Replace "5" with "X", the number of minutes between two temperature readings.

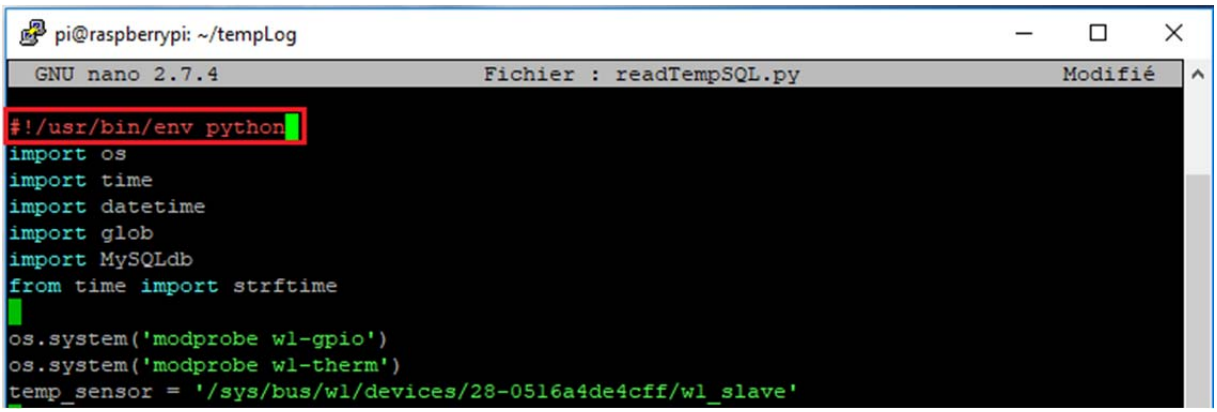


As it stands, auto-registration will not work because the script "readTempSQL.py" is not yet executable and the Cronjob will fail.

```
cd /home/pi/tempLog
sudo nano readTempSQL.py
```

To make the script executable, you must first add a line to the "readTempSQL.py" file:

```
#!/usr/bin/env python
```



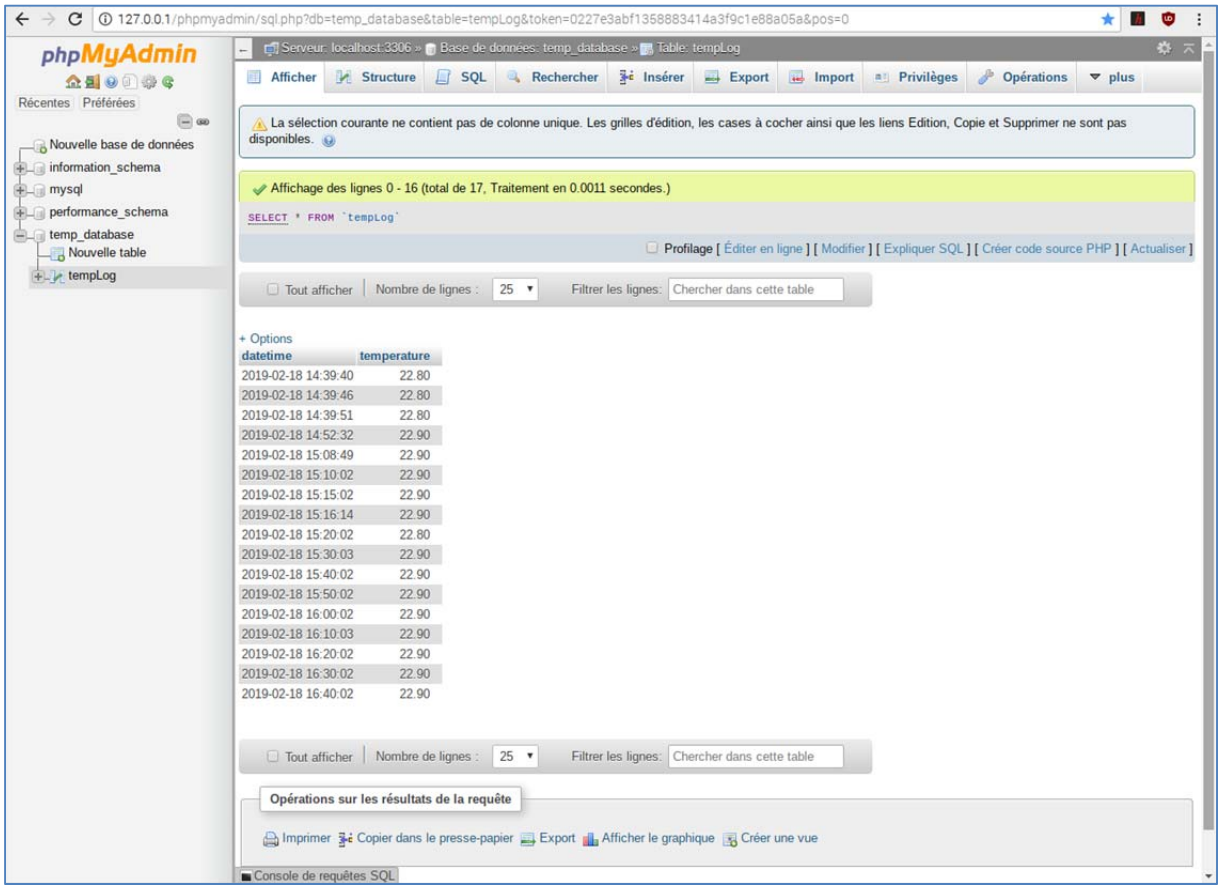
Then, you need to change the authorization of the file "readTempSQL.py" in order to make it executable ("+x" on Linux):

```
sudo chmod +x readTempSQL.py
```

To test that the file is now executable, navigate to your "tempLog" directory and run:

```
./readTempSQL.py
```

The file is now executable. Check your database on phpMyAdmin after 10 minutes to make sure your Cronjob is working.



3.7. Exporting data in a JSON file

Let's create a PHP script to export the data from our database to a file. Go to the directory "/var/www/html" and create a new file "temperaturejson.php":

```
cd /var/www/html
```

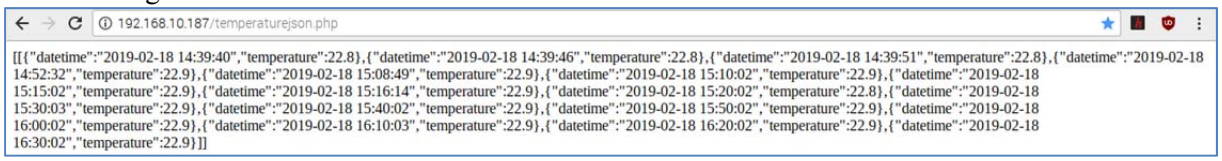
```
sudo nano temperaturejson.php
```

```
<?php

//STORE INFOS INTO VARIABLES
$host    = 'localhost';
// SQL DB NAME
$db      = 'temp_database';
// CONNECT TO THE DB
$user    = 'root';
$pass    = 'password';
$charset = 'utf8';

// LOGIN PROTOCOL USING PDO FORMAT
$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$opt = [
        PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
        PDO::ATTR_EMULATE_PREPARES   => false,
        ];

$dbh = new PDO($dsn, $user, $pass, $opt);

// SQL QUERY TO READ ALL DATA FROM THE tempLog TABLE
$sql  = $dbh->query("SELECT * FROM tempLog");
$rows = array();
while ($row = $sql->fetchall()) {
    $rows[] = $row;
}
//RETURN DATA IN JSON FORMAT
echo json_encode($rows);

?>
```

This PHP script retrieves all the data from the "tempLog" table from our "temp_database" database with the query "SELECT * FROM tempLog" and writes them into a table each time a set of "datetime" and "temperature" data is read. This aggregator table is then encoded in JSON and displayed using "echo".

To run the script and display its result, from your browser go to "http://hostanme/temperaturejson.php". Use your own IP address. You can obtain it with the command "hostame -I"

You should get a similar result:



4. How to monitor a temperature with Zabbix and Grafana

This third part describes the steps for installing, configuring and interfacing Zabbix and Grafana on Raspbian to monitor a temperature.

4.1. Framework

We now want to monitor and graphically display our temperature readings measured by a DS18B20 probe connected to a Raspberry Pi.

The readings are, at this stage, registered every 10 minutes in a MySQL database hosted by our Raspberry Pi on which we have installed and configured a LAMP server.

To carry out this monitoring, we will use the Zabbix software and its components. We will also use Grafana software for the graphical quality of its dashboards on which we will import several types of data sources (Zabbix, MySQL, etc.). An interface between Zabbix and Grafana must be configured.

Zabbix is a free software that monitors the status of various network services, servers and other network hardware and produces dynamic resource consumption graphs.

Grafana is a free software under Apache 2.0 license that allows the visualization and formatting of metric data. It allows you to create dashboards and graphs from several sources including time series databases such as Graphite, InfluxDB and OpenTSDB.

4.2. Installing and configuring Zabbix

4.2.1. Installing Zabbix and its components

- **Installing Zabbix**

Run the following commands to install Zabbix from its repository:

```
sudo wget https://repo.zabbix.com/zabbix/4.0/raspbian/pool/main/z/zabbix-release/zabbix-release_4.0-2+stretch_all.deb
```

```
sudo dpkg -i zabbix-release_4.0-2+stretch_all.deb
```

```
sudo apt update
```

```
sudo apt upgrade
```

- **Installing components**

Install the server, frontend, agent components for Zabbix:

```
sudo apt -y install zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

Nota Bene:
In this procedure, the Zabbix server and Zabbix agent components will be installed on the same Raspberry Pi. If the server and the Zabbix agent are running on the same machine, it is recommended to use a different user to run the server and the agent. Otherwise, if both are executed by the same user, the agent can access the server configuration file and any administrator-level user in Zabbix can very easily recover, for example, the database password.

- **Créer une base de données pour Zabbix**

Access MySQL:

```
mysql -uroot -p
```

Enter your « *root* » password.

Run the following MySQL commands to create a database named zabbix and grant access privileges to the zabbix user. The zabbix user was automatically created during the installation of Zabbix:

```
CREATE DATABASE zabbix CHARACTER SET utf8 COLLATE utf8_bin;
```

```
GRANT ALL PRIVILEGES ON zabbix.* TO zabbix@localhost IDENTIFIED BY 'password';
```

```
QUIT;
```

- **Manage user privileges for MySQL (optional)**

In case the command to grant privileges to the user "zabbix" by "root" does not work and displays an error message (see capture below) you can perform the procedure described in this section.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 51
Server version: 10.1.37-MariaDB-0+deb9ul Raspbian 9.0

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost identifie
d by 'password';
ERROR 1044 (42000): Access denied for user 'root'@'localhost' to database 'zabbi
x'
MariaDB [(none)]>
```

Force MySQL update:

```
sudo mysql_upgrade -u root -p --force
```

Access MySQL:

```
mysql -uroot –p
```

Check that the "root" user is able to grant privileges to other users:

```
SELECT * FROM mysql.user WHERE User='root'\G;
```

```
MariaDB [(none)]> SELECT * FROM mysql.user WHERE User='root'\G;
*************************** 1. row ***************************
                Host: localhost
                User: root
            Password: *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
          Select_priv: Y
          Insert_priv: Y
          Update_priv: Y
          Delete_priv: Y
          Create_priv: Y
            Drop_priv: Y
          Reload_priv: Y
        Shutdown_priv: Y
         Process_priv: Y
            File_priv: Y
           Grant_priv: N
      References_priv: Y
           Index_priv: Y
           Alter_priv: Y
         Show_db_priv: Y
           Super_priv: Y
Create_tmp_table_priv: Y
     Lock_tables_priv: Y
         Execute_priv: Y
      Repl_slave_priv: Y
     Repl_client_priv: Y
     Create_view_priv: Y
       Show_view_priv: Y
  Create_routine_priv: Y
   Alter_routine_priv: Y
     Create_user_priv: Y
```

Alternatively, execute the SQL command displaying the privilege status for all users:

```
SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;
```

```
MariaDB [(none)]> SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;
+-----------+--------+-------------------------------------------+------------+------------+
| host      | user   | password                                  | Grant_priv | Super_priv |
+-----------+--------+-------------------------------------------+------------+------------+
| localhost | root   | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N          | Y          |
| %         | zabbix | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N          | N          |
+-----------+--------+-------------------------------------------+------------+------------+
2 rows in set (0.00 sec)
```

On the two previous screenshots, we notice that the "root" administrator user is not able to grant access privileges to other users (Super_priv: N).

Run the following SQL commands to fix this:

```
UPDATE mysql.user SET Grant_priv='Y', Super_priv='Y' WHERE User='root';
FLUSH PRIVILEGES
SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;
```

Then, check again. It may be necessary to exit and restart MySQL to ensure that the new configuration is properly taken into account.

```
MariaDB [(none)]> UPDATE mysql.user SET Grant_priv='Y', Super_priv='Y' WHERE User='root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [(none)]> SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;
+-----------+--------+-------------------------------------------+------------+------------+
| host      | user   | password                                  | Grant_priv | Super_priv |
+-----------+--------+-------------------------------------------+------------+------------+
| localhost | root   | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | Y          | Y          |
| %         | zabbix | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N          | N          |
+-----------+--------+-------------------------------------------+------------+------------+
2 rows in set (0.00 sec)
```

To ensure that the root user has full privileges on all MySQL databases (" *") and covering all host types (" %"'), run the SQL command:

```
GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'password';
```

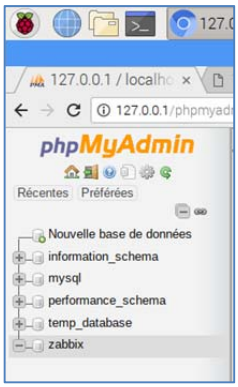
Restart MySQL and run:

```
GRANT ALL ON zabbix.* TO zabbix@localhost IDENTIFIED BY 'password';
```

- **Checking the database**

You can check the creation of the "zabbix" database with the SQL command "SHOW DATABASES;" or directly in the phpMyAdmin GUI at the address "127.0.0.1/phpmyadmin/".

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| temp_database      |
| zabbix             |
+--------------------+
5 rows in set (0.01 sec)
```



- **Importing initial schema and data**

For "zabbix" user:

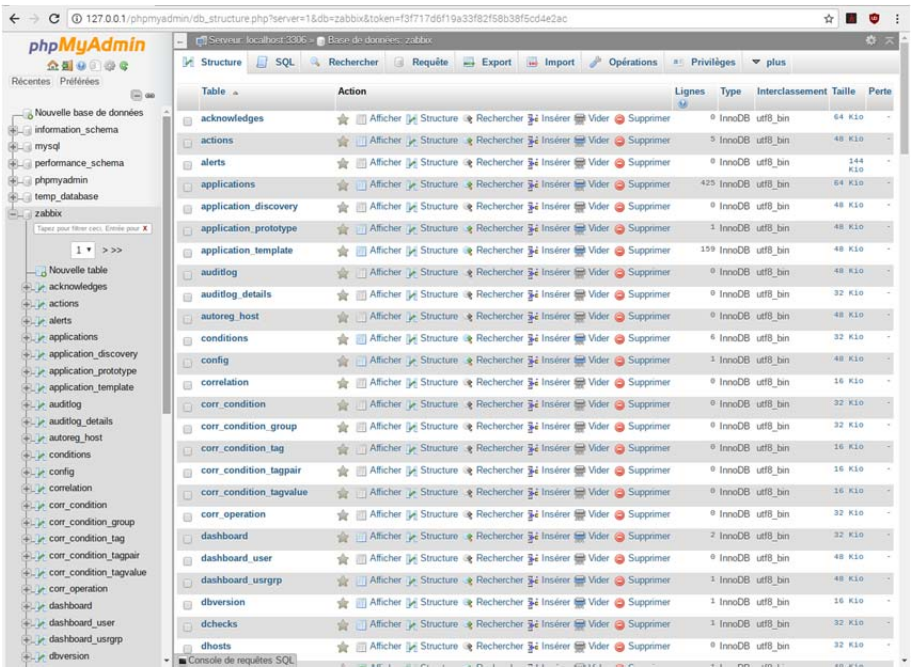
```
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

For "root" user:

```
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uroot -p zabbix
```

Enter your password.

Check if the import was successful on phpMyAdmin:



4.2.2. Configuring Zabbix

- **Configuring the database**

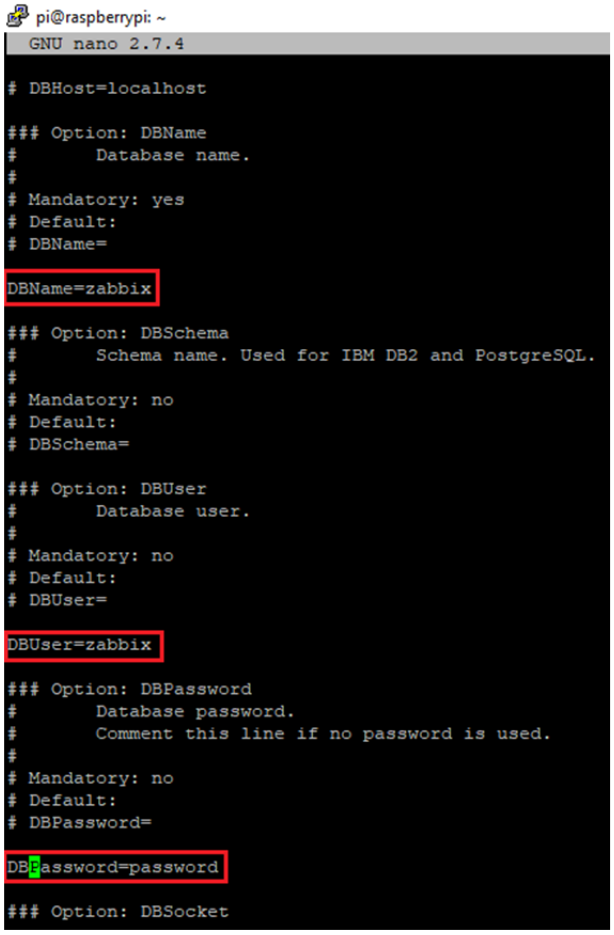
Edit the file "/etc/zabbix/zabbix_server.conf":

```
sudo nano /etc/zabbix/zabbix_server.conf
```

Add the password corresponding to your database at:

```
DBPassword=password
```

Check the database name, user name, port. You can change the cache size so that the server is able to support a larger number of hosts:




```
DBName = zabbix

DBUser=zabbix

ListenPort=10051

CacheSize=32
```

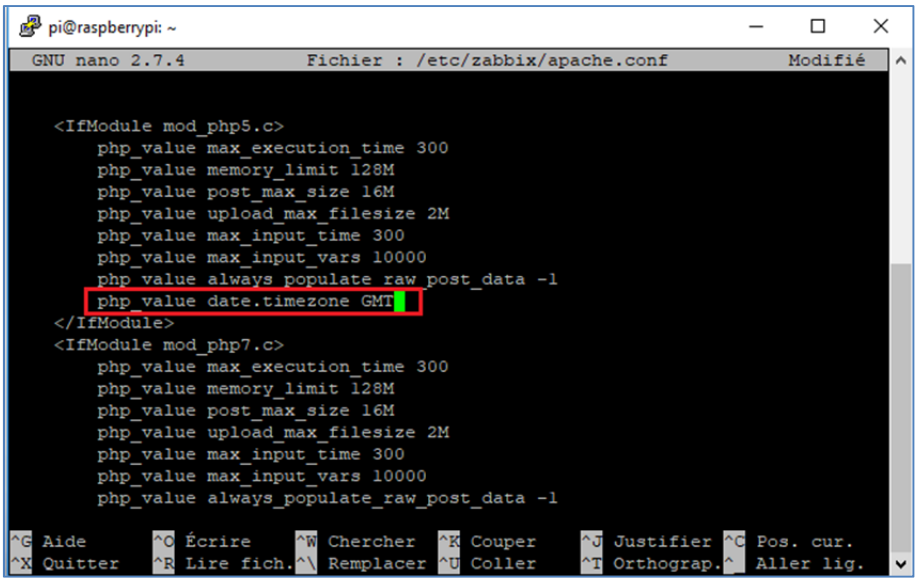
- **Configuring PHP**

Edit the file "/etc/zabbix/apache.conf":

```
sudo nano /etc/zabbix/apache.conf
```

Uncomment the lines below and enter your time zone for versions 5 and 7 of the PHP module (Paris time: UTC+0100):

```
php_value date.timezone Europe/Paris
```



- Starting Zabbix

Start Zabbix server and Zabbix agent:

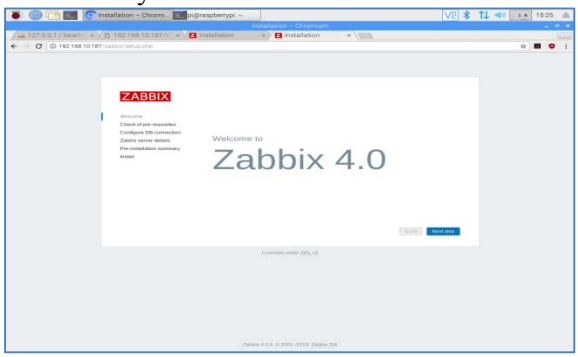
```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

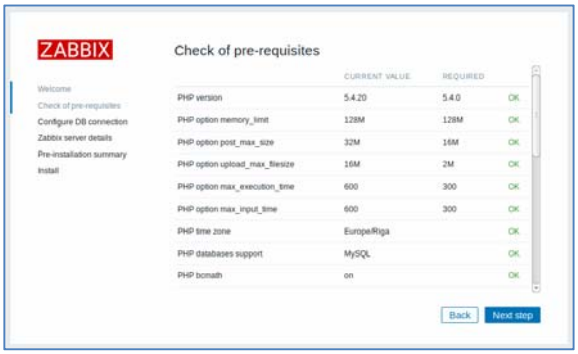
Make them start automatically at system startup

```
sudo systemctl enable zabbix-server zabbix-agent apache2
```

4.2.3. Installing Zabbix front-end

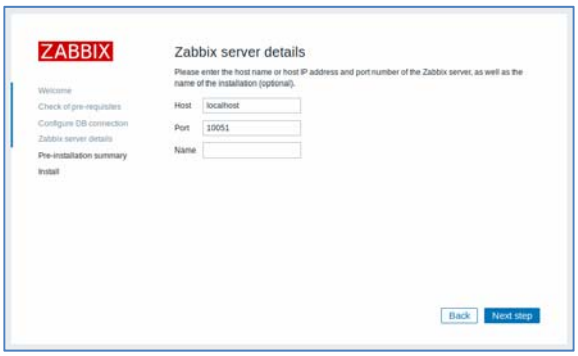
Connect to your Zabbix frontend server at the address "localhost/zabbix/":





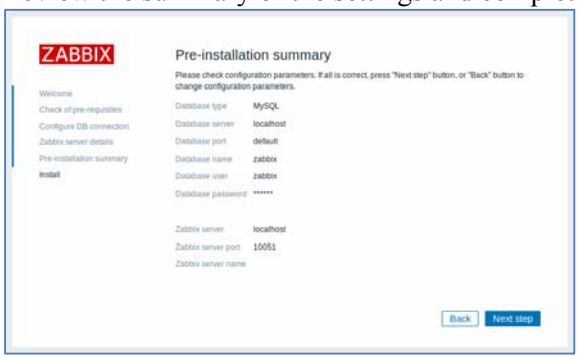
Then, make sure that all software requirements are met.

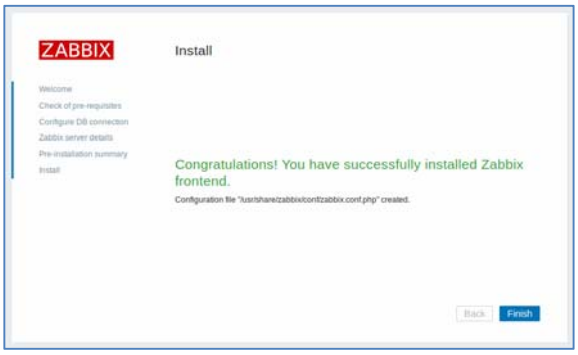
Enter the connection details to the "zabbix" database:



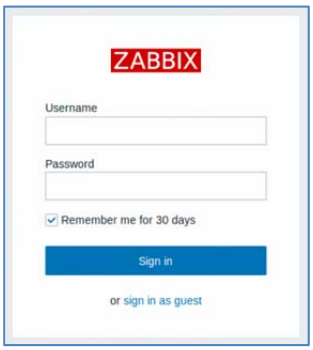
Entering a name for the Zabbix server is optional

Review the summary of the settings and complete the installation.



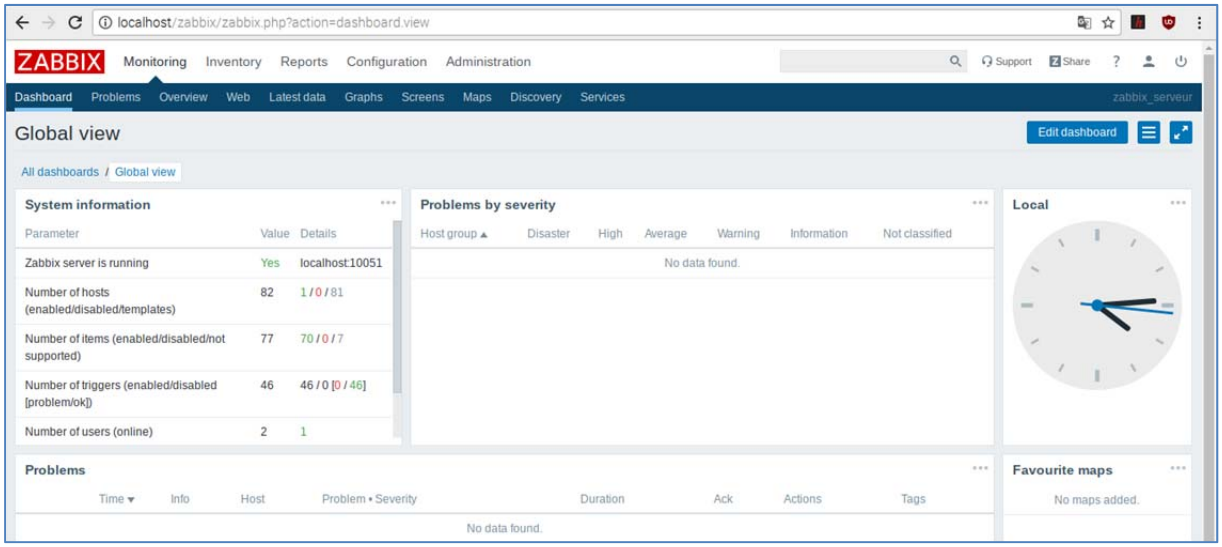


Zabbix front-end is ready. The default user name is "Admin" and the password is "zabbix".

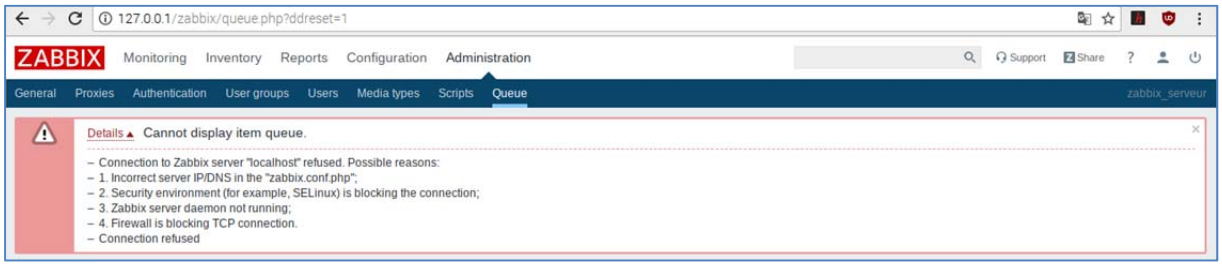


4.2.4. Configuring and managing Zabbix front-end

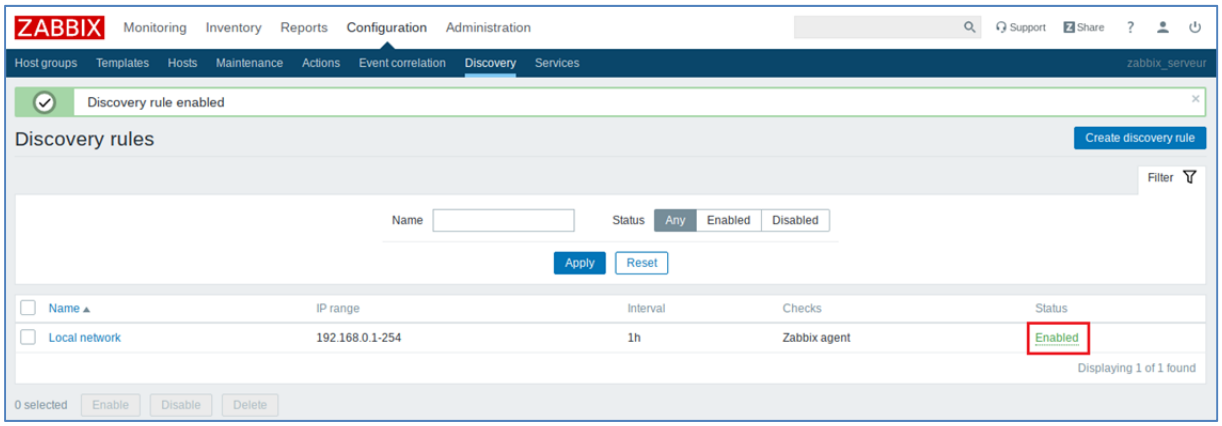
The connection with the server is operational on port 10051 if you observe "Zabbix server is running" with "Value:Yes".



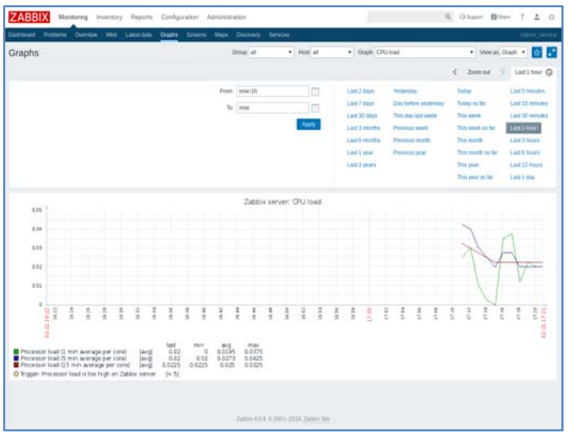
If you see an error when connecting to Zabbix Front-end (see capture below) check the configuration of the file "/etc/zabbix/zabbix/zabbix_server.conf" or the user rights in MySQL with the SQL command: "SELECT host,user,password,Grant_priv,Super_priv FROM mysql.user;" by referring to the appropriate above sections.



Also check that your discovery rules are enabled for your local network in the menu "Configuration > Discovery > Status: Enabled".



Preconfigured items are immediately available in the "Monitoring > Graphs" tab such as the processor load curve of the Raspberry Pi that can be viewed on the right.



4.3. How to display a temperature in zabbix

We will now configure Zabbix to collect, log and display a temperature.

From a simple command returning a measurement of the DS18B20 "28-*" sensor:

```
sudo /bin/cat /sys/bus/w1/devices/28-0516a4de4cff/w1_slave | awk '/t=/ {print $10}' | cut -d= -f2 | awk '{sum=$1/1000} END {print sum}'
```

Go to the folder "etc/zabbix":

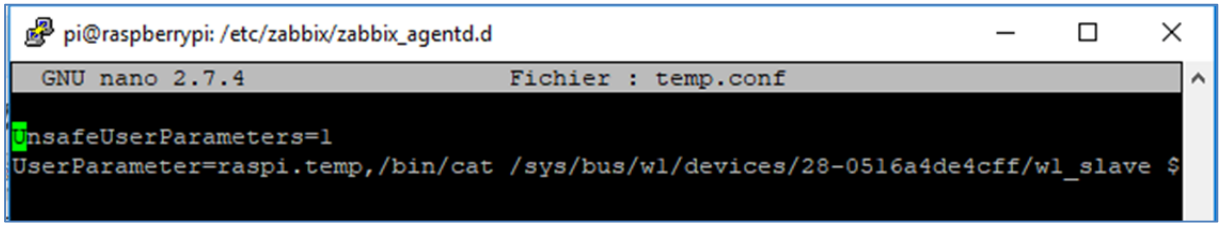
```
cd /etc/zabbix/zabbix_agentd.d
```

Create and edit a new file "temp.conf":

```
sudo nano temp.conf
```

Add the command to the configuration file that will store the information in Zabbix to display it:

```
UnsafeUserParameters=1

UserParameter=raspi.temp,/bin/cat /sys/bus/w1/devices/28-0516a4de4cff/w1_slave | awk '/t=/ {print $10}' | cut -d= -f2 | awk '{sum=$1/1000} END {print sum}'
```



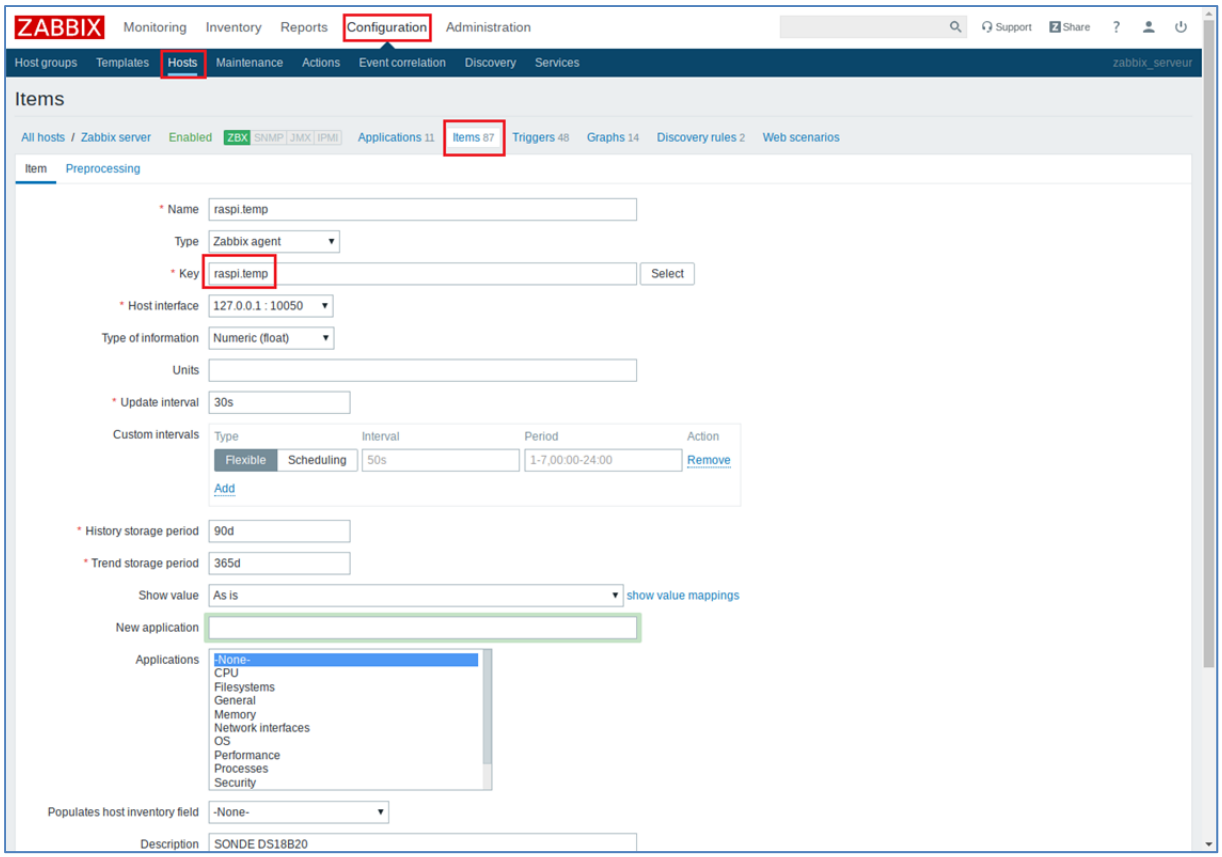
Finally, restart the Zabbix services:

```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

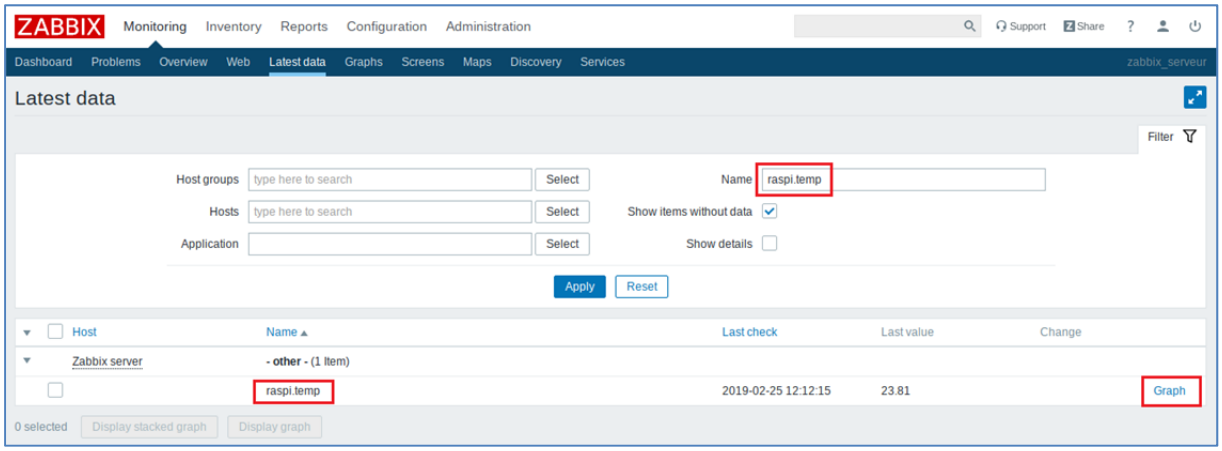
Under Zabbix Agent, go to "Configuration > Hosts > Items > Create new item", and enter the name "Name" and the key "Key" in correspondence with the name of the chosen variable of "UserParameter" in the file previously created "temp.conf", namely "raspi.temp".

For the "Host Interface" box, enter the IP address of the host you want to monitor. The agent port is by default "10050" for Zabbix Agent (10051 for Zabbix Server).

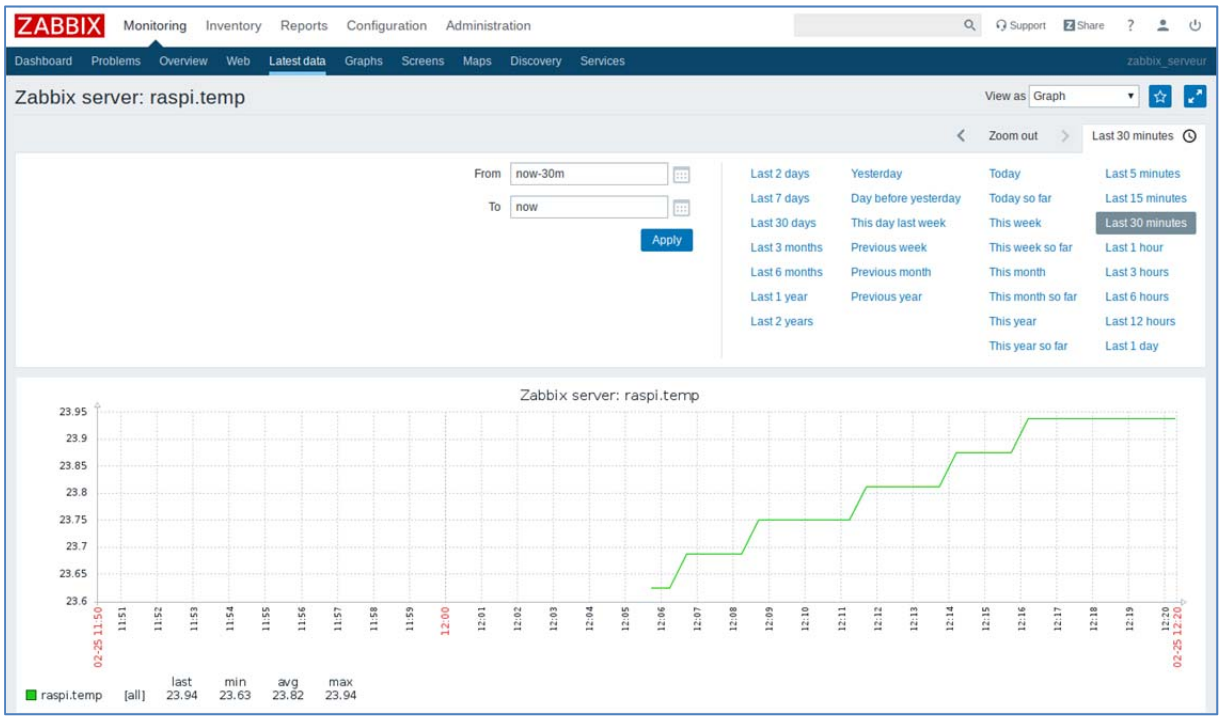
You can also configure the data collection interval with "Update interval" and its storage time in the database with "History storage period".



Then go to the menu "Monitoring > Latest data", filter on the name of your object, namely "raspi.temp", you should see a data under "Last value".



Click on "Graph" to view the temperature measurements:



4.4. Installing Grafana and Zabbix plugin

4.4.1. Installing Grafana

Get Grafana's sources:

```
sudo apt-get install apt-transport-https curl
```

```
curl https://bintray.com/user/downloadSubjectPublicKey?username=bintray | sudo apt-key add –
```

```
echo "deb https://dl.bintray.com/fg2it/deb stretch main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

```
sudo apt-get update
```

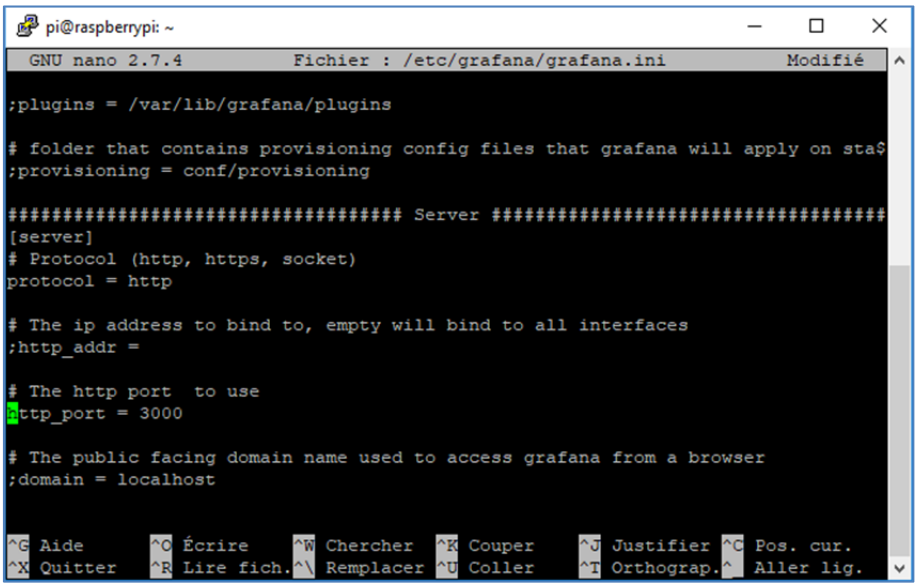
```
sudo apt-get install grafana
```

For security reasons, the HTTP protocol that allows access to the WEB interface to the Grafana API is disabled. To enable the HTTP protocol, modify the file "/etc/grafana/grafana/grafana.ini":

```
sudo nano /etc/grafana/grafana.ini
```

Uncomment the http protocol and port (remove the ";"):

```
[server]
# Protocol (http, https, socket)
protocol = http

# The ip address to bind to, empty will bind to all interfaces
;http_addr =

# The http port  to use
http_port = 3000
```



Restart Grafana:

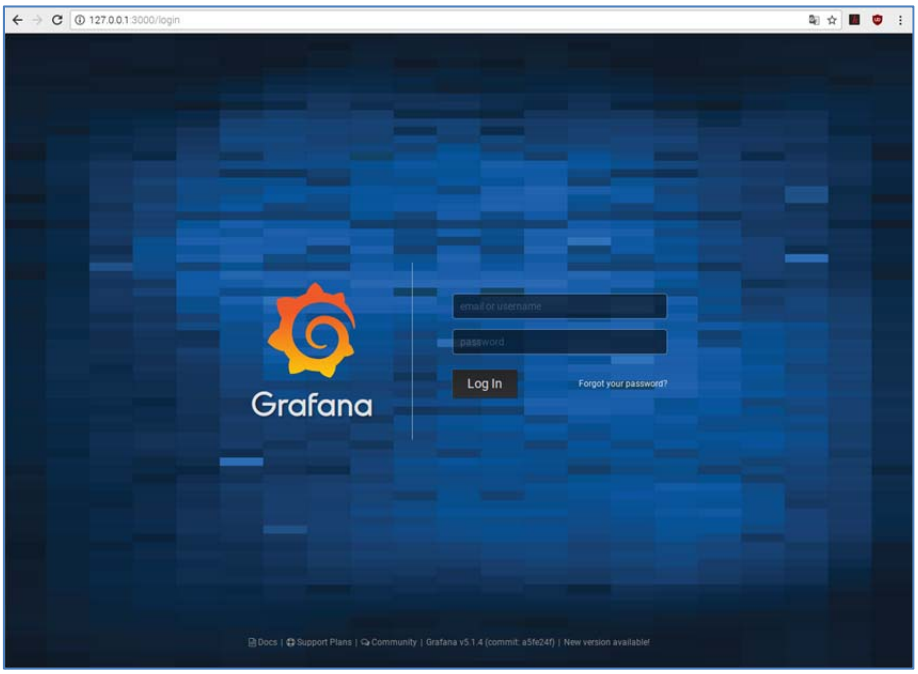
```
sudo service grafana-server restart
```

Enable the "systemd" service so that Grafana starts when the system starts:

```
sudo systemctl enable grafana-server.service
```

4.4.2. Connecting to Grafana

Open your browser and enter the address "http://localhost:3000/login". The default credentials are "admin" / "admin". Note that Grafana has a user management with authentication and rights management.



4.4.3. Installing Grafana-Zabbix plugin

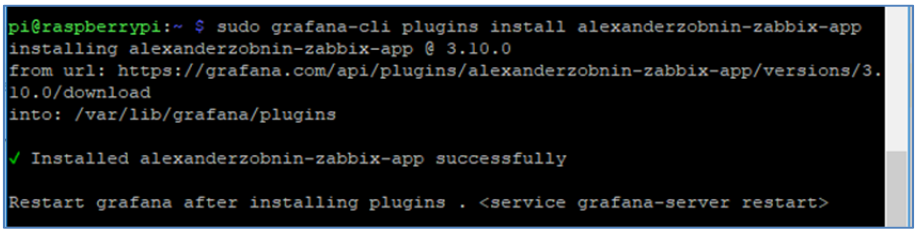
You can get a list of the plugins available for Grafana in your terminal:

```
grafana-cli plugins list-remote
```

Install the Zabbix module with grafana-cli:

```
grafana-cli plugins install alexanderzobnin-zabbix-app
```

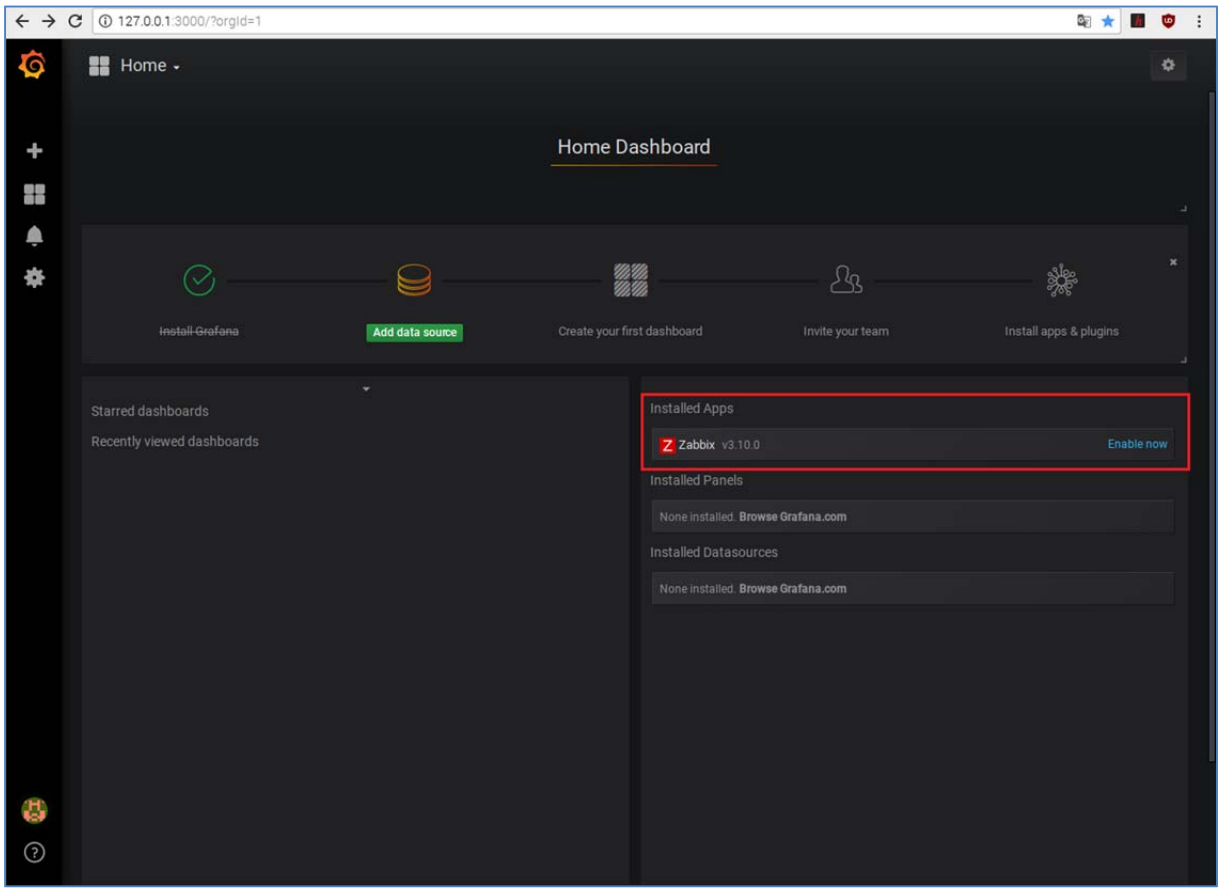
The plugin will be installed in your Grafana plugin directory. The default path is "/var/lib/grafana/plugins".



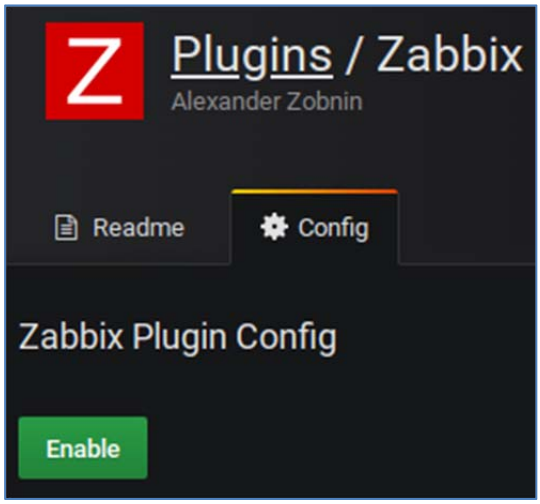
Restart Grafana:

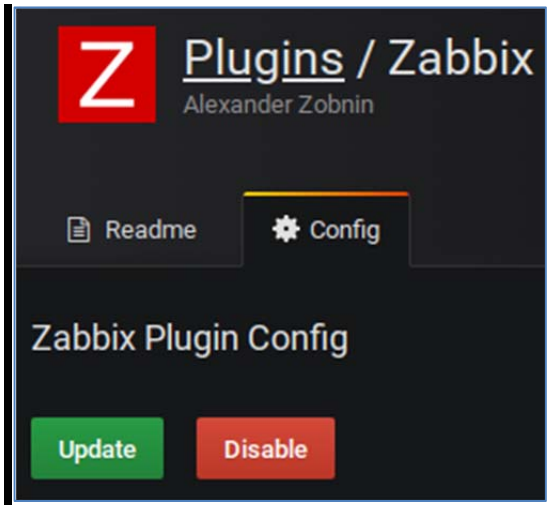
```
service grafana-server restart
```

You will see on the Grafana homepage the appearance of a Zabbix icon under the right side panel "Installed Apps".



To activate the Zabbix module click on "Enable Now", then on "Enable". You should see the icons change to "Update" and "Disable".



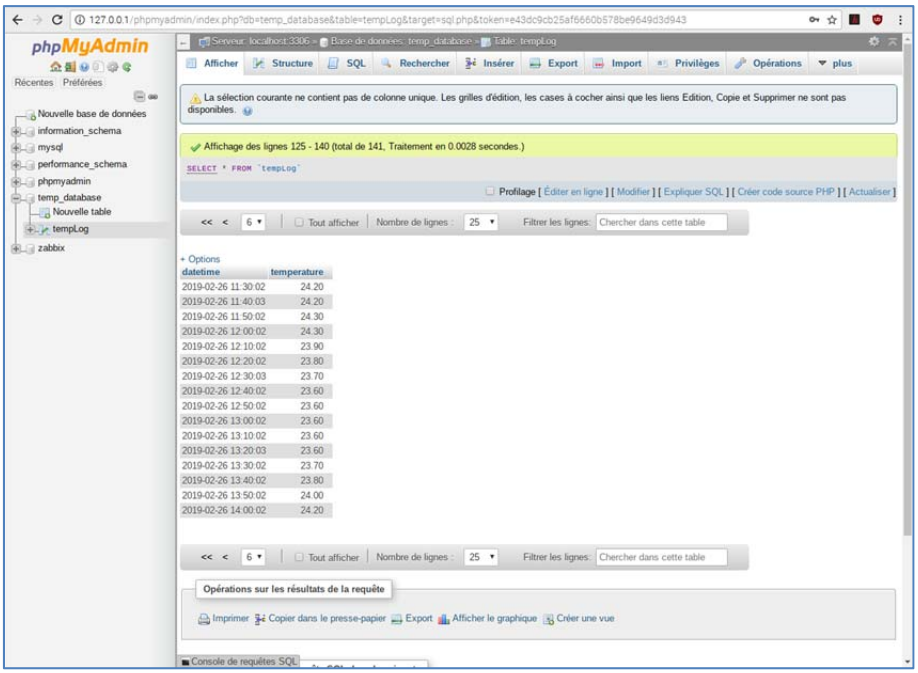


4.5. Displaying a temperature in Grafana

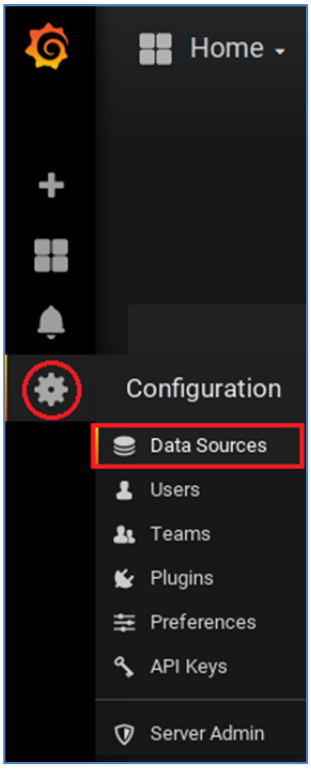
4.5.1. Importing a MySQL source on Grafana

- **Configuring the import**

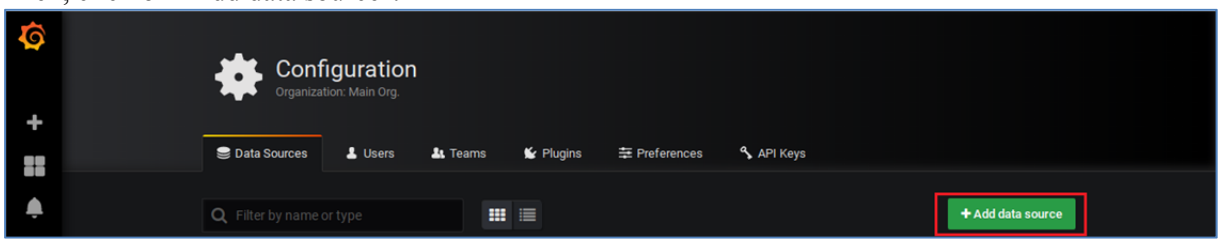
We will import these measurements from our SQL "tempLog" into Grafana and create a custom dashboard with two panels, one with raw data, the other graphical.



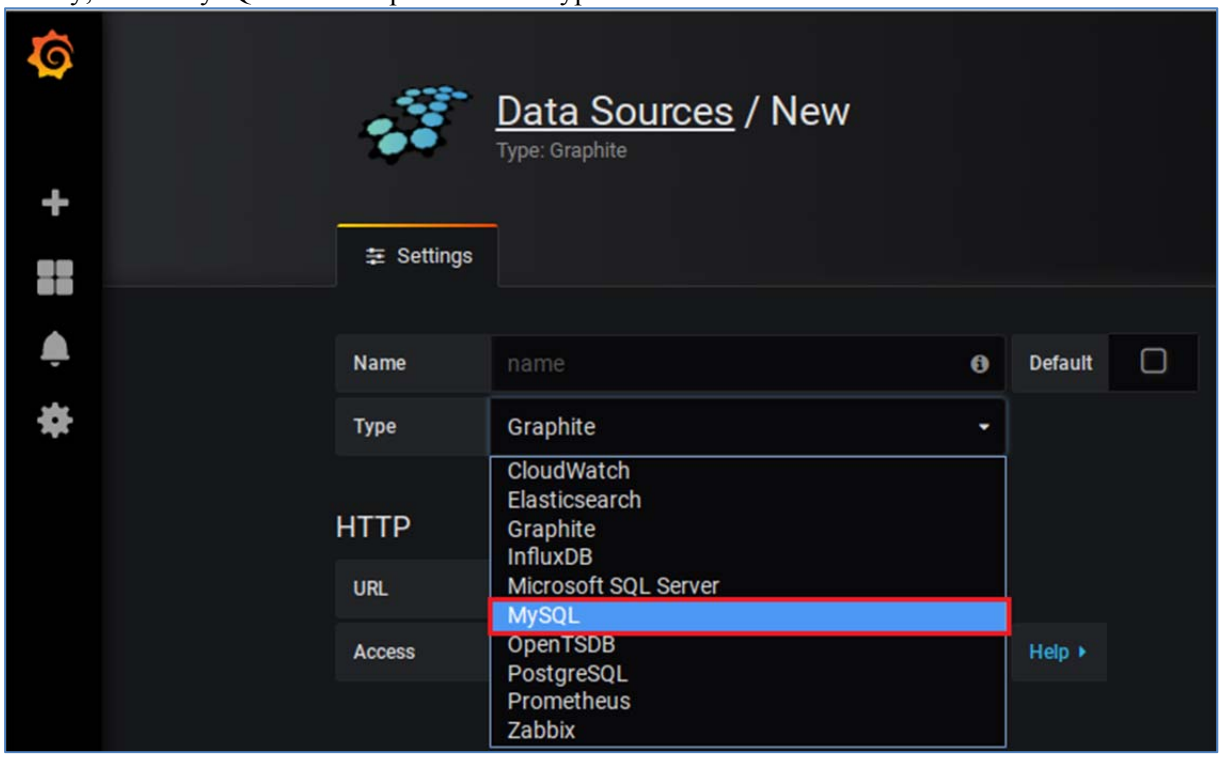
To add a new MySQL data source, open "Configuration" in the side panel and click on "Data Sources".



Then, click on "Add data source".

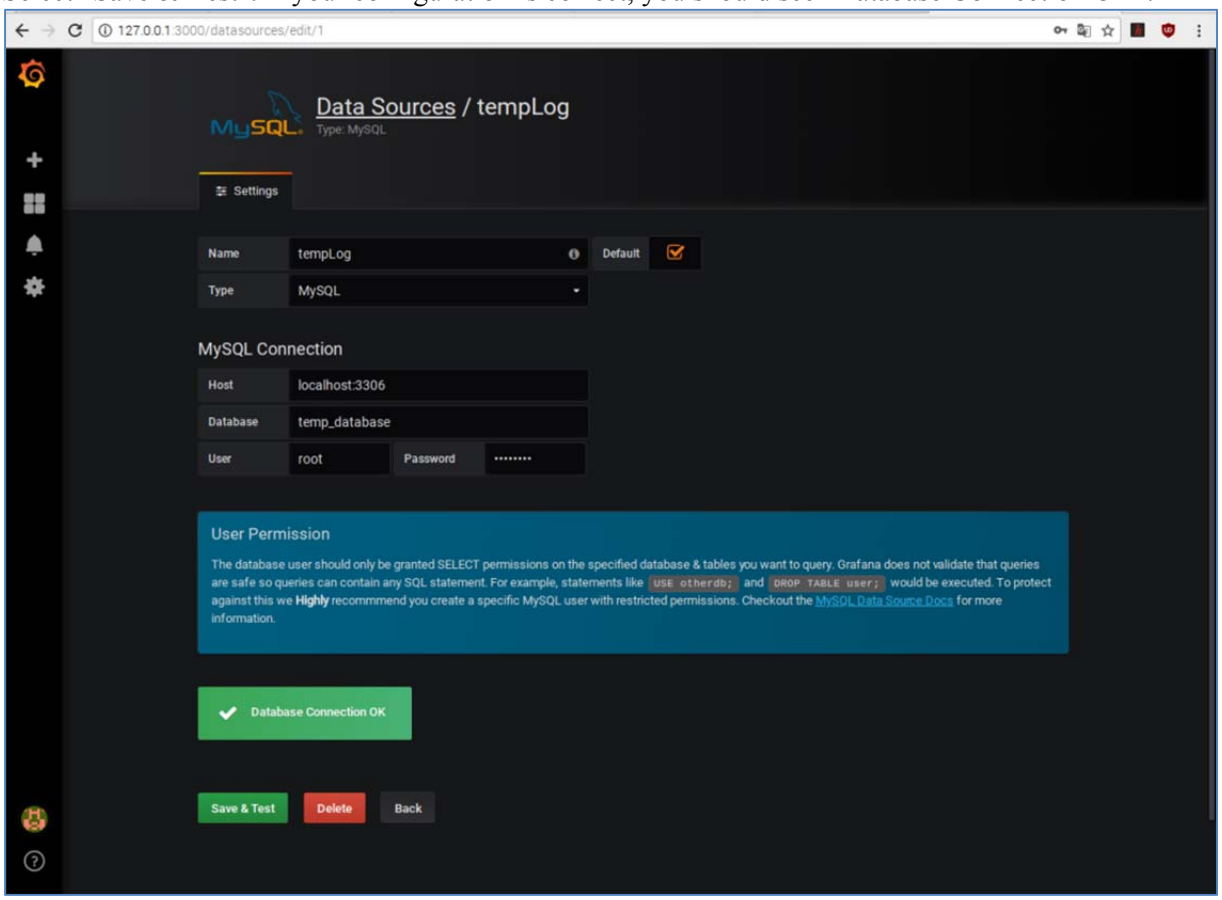


Finally, select MySQL in the drop-down list "Type".

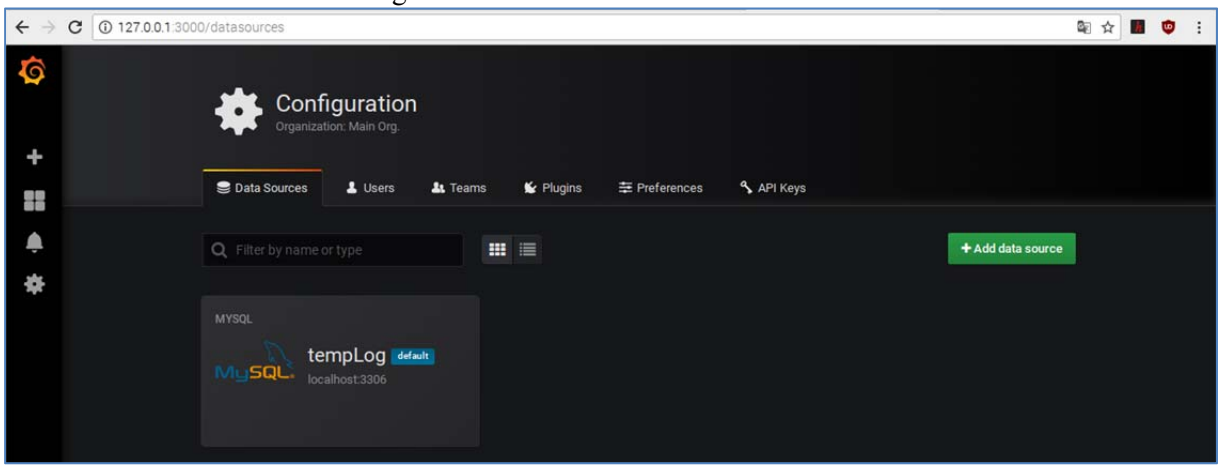


Configure the MySQL source named "tempLog" by entering the name of the MySQL database ("temp_database") to import, and the access credentials to this database with User ("root") and Password ("password"). The "Host" port for MySQL is by default "3306".

Select "Save & Test". If your configuration is correct, you should see "Database Connection OK".



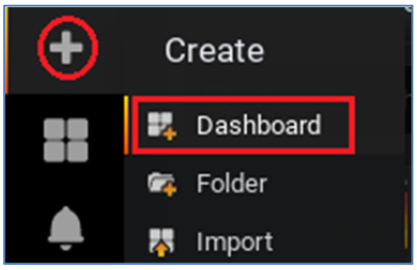
The source is saved in the "Configuration" menu:

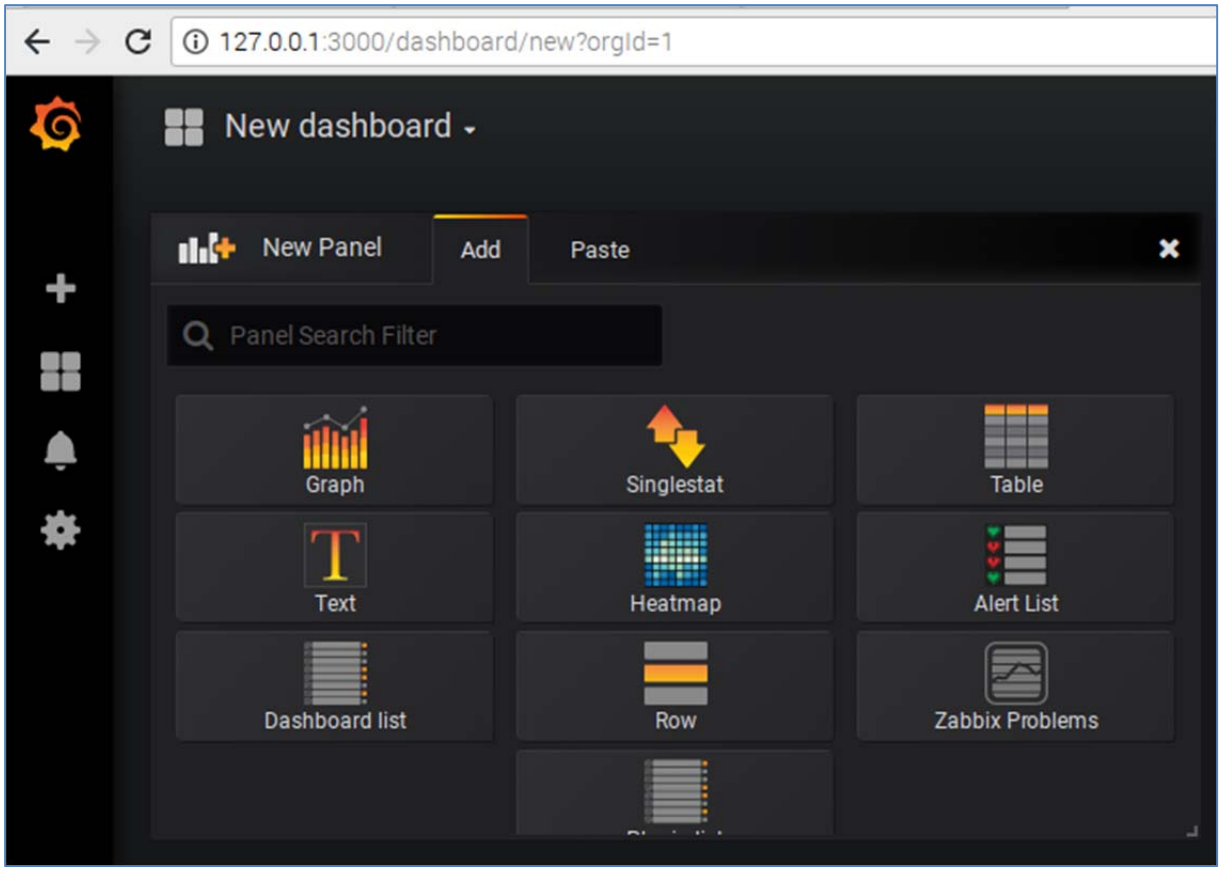


Now that we have configured our MySQL source, we can create a Dashboard, which we will call "Temperature - Server Room (DS18B20)", and which will display our temperatures.

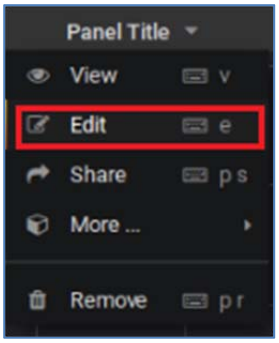
- **Creating a "Table" panel**

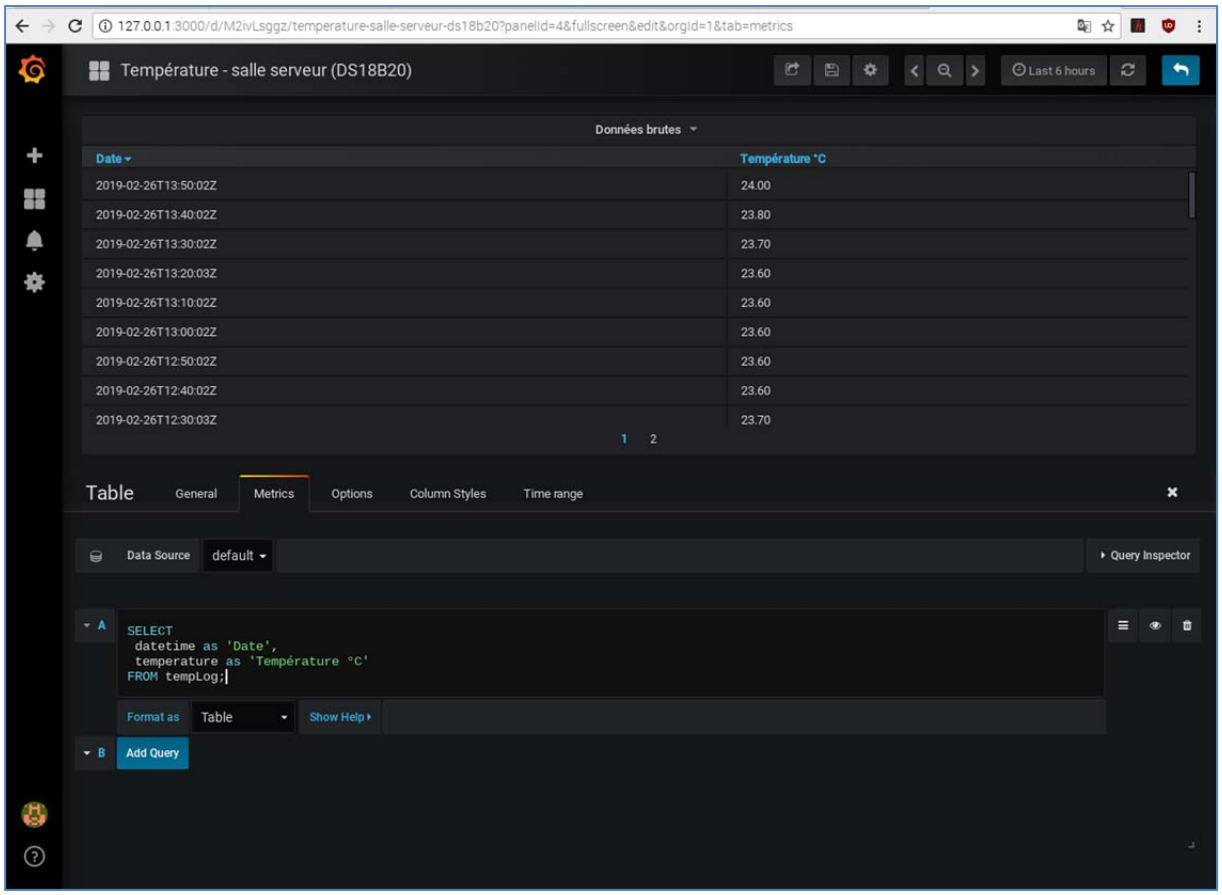
Open the "Create" menu in the side panel and click on "Dashboard" in the drop-down list.
To display the raw information collected in the MySQL table "tempLog" and display the values of the columns "datetime" and "temperature", we will create a panel of type "Table" and edit it in our "New dashboard".





Once the "Table" panel has been created, to edit it, select "Edit" from the drop-down menu next to "Panel Title".
The name of the panel can be modified in the "General" tab.





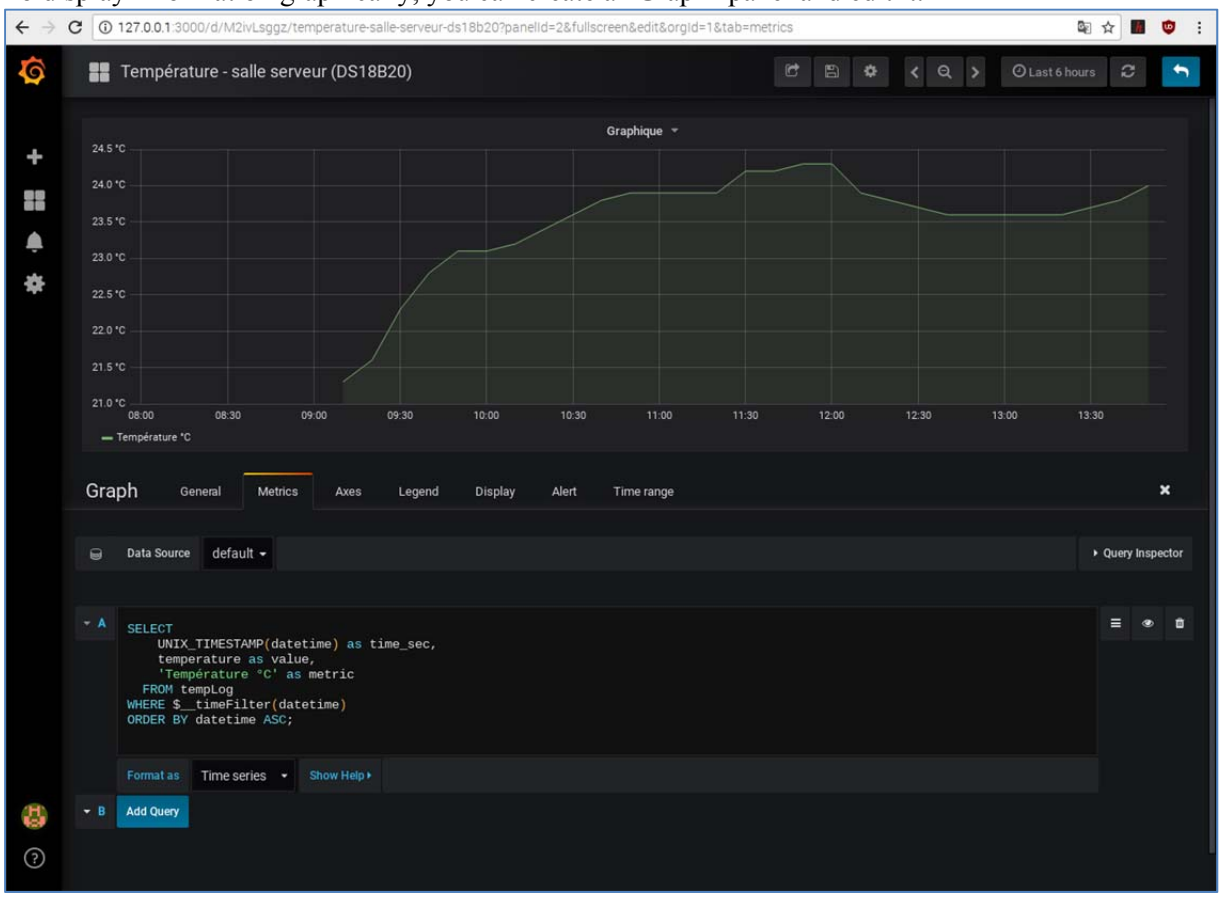
In the "Metrics" tab, enter the following SQL query:

```
SELECT
   datetime as ‘Date’,
   temperature as ‘Température °C’
FROM tempLog;
```

If you want to change the name of a column for a more accurate display, it is necessary to enter it in the SQL query (e. g. "temperature as 'Temperature °C' ").

- **Creating a "Graph" panel**

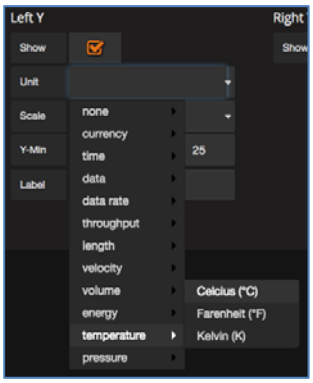
To display information graphically, you can create a "Graph" panel and edit it.



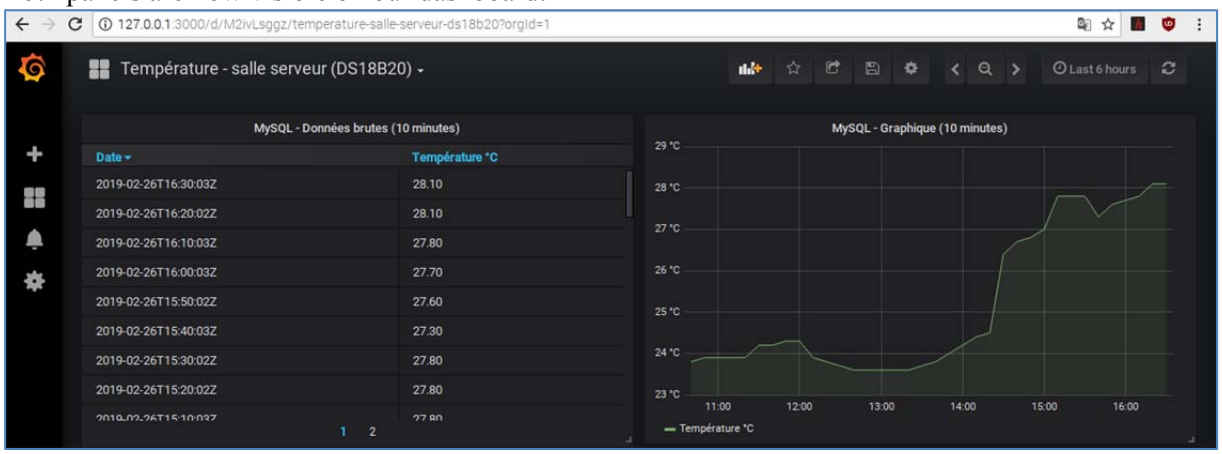
In the "Metrics" tab, enter the following SQL query:

```
SELECT
  UNIX_TIMESTAMP(datetime) as time_sec,
  temperature as value,
  'temperature' as metric
 FROM tempLog
WHERE $__timeFilter(datetime)
ORDER BY datetime ASC;
```

To change the y-axis unit and display a legend in °Celsius, choose from the Axes tab: "Axes > Left Y > Unit > temperature > Celsius (°C)".



Both panels are now visible on our dashboard:



4.5.2. Importing a Zabbix source on Grafana

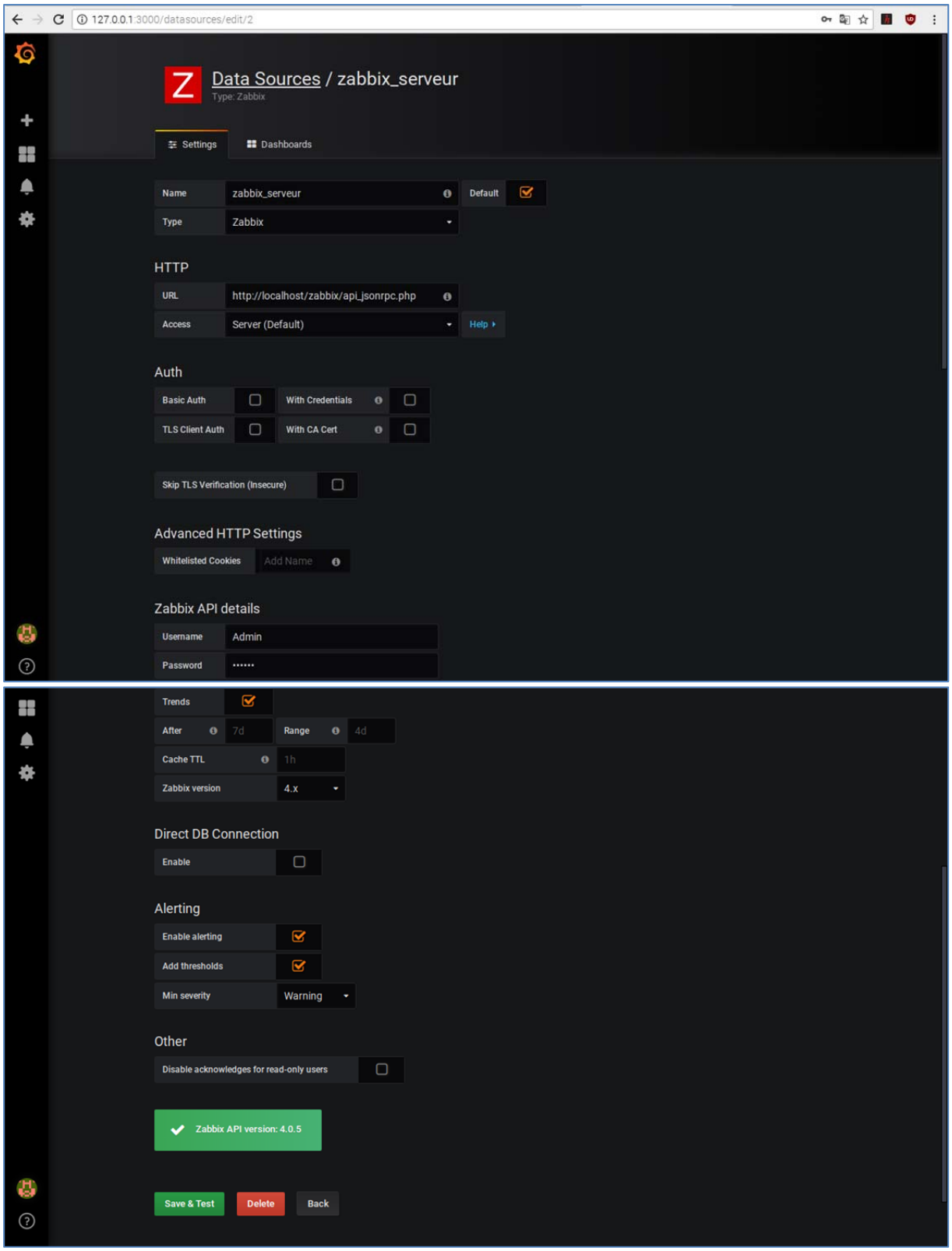
- **Configuring the import**

After activating the plugin, you can add the Zabbix data source. To add a new MySQL data source, open "Configuration" in the side panel and click on "Data Sources". Then click on "Add data source". Finally, select "MySQL" from the "Type" drop-down list.

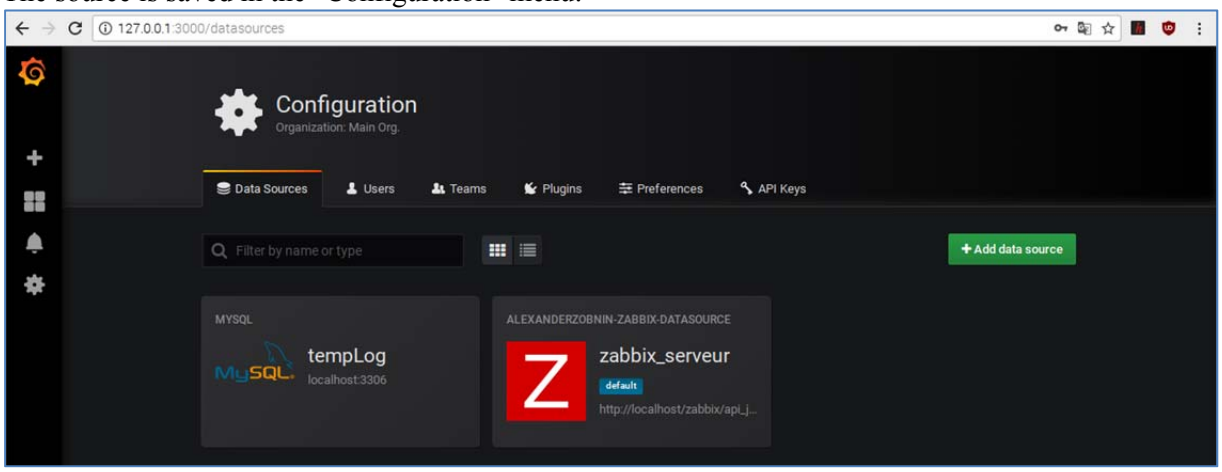
The important fields to be filled in are:

- Name: The name of the server. Here named after our Zabbix server "zabbix_server";
- Host: Fill in the URL of the full path to Zabbix with "api_jsonrpc.php". Here: "http://localhost/zabbix/api_jsonrpc.php";
- Access: Left by default in "Server (Default)";
- Userame and Password: You must enter your Zabbix login details (the default login details are "Admin" / "zabbix").

For more information on configuring a Zabbix source in Grafana, please consult the appropriate documentation: http://docs.grafana-zabbix.org/installation/configuration/.


127.0.0.1:3000/datasources/edit/2
Data Sources / zabbix_serveur
Type: Zabbix
Settings
Dashboards
Name
zabbix_serveur
Default
Type
Zabbix
HTTP
URL
http://localhost/zabbix/api_jsonrpc.php
Access
Server (Default)
Help
Auth
Basic Auth
With Credentials
TLS Client Auth
With CA Cert
Skip TLS Verification (Insecure)
Advanced HTTP Settings
Whitelisted Cookies
Add Name
Zabbix API details
Username
Admin
Password
Trends
After
7d
Range
4d
Cache TTL
1h
Zabbix version
4.x
Direct DB Connection
Enable
Alerting
Enable alerting
Add thresholds
Min severity
Warning
Other
Disable acknowledges for read-only users
Zabbix API version: 4.0.5
Save & Test
Delete
Back

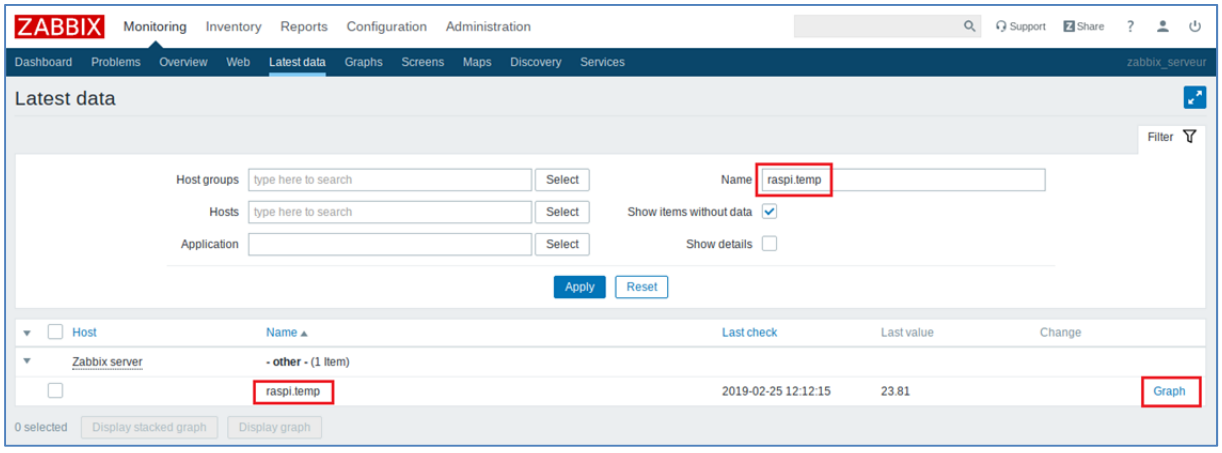
The source is saved in the "Configuration" menu:



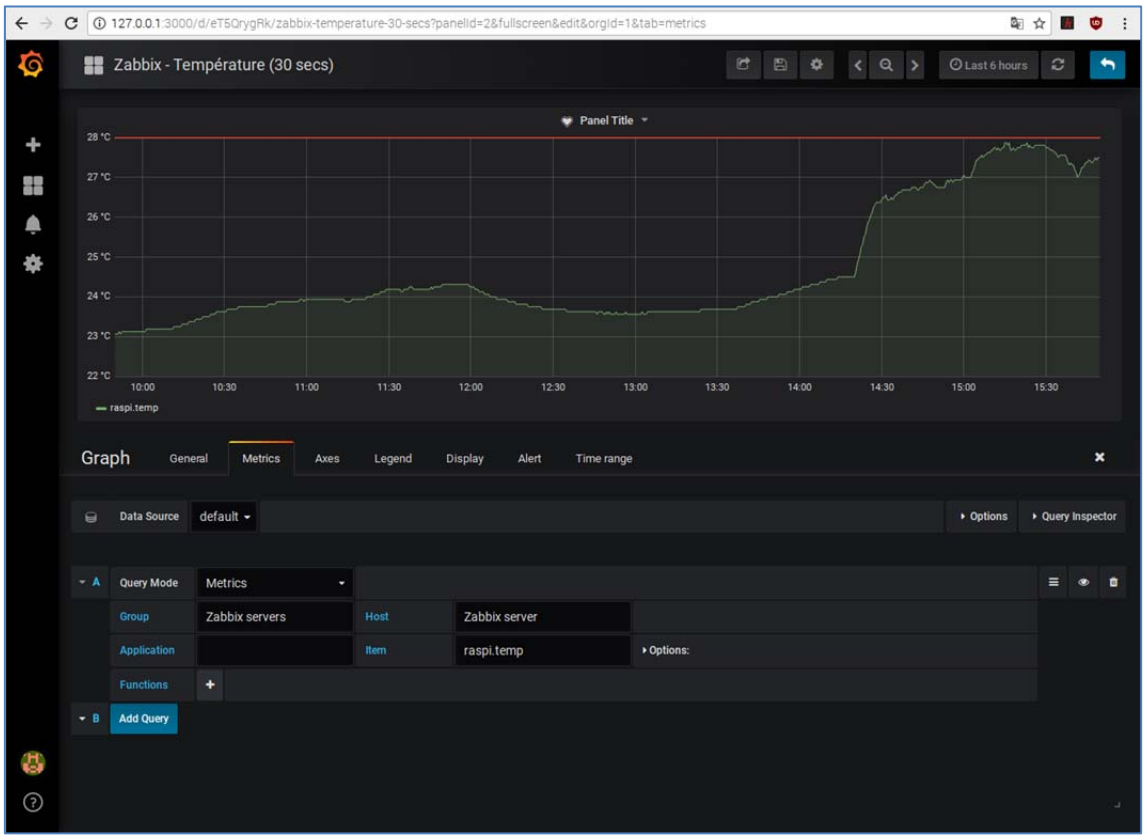
- **Creating a Dashboard with a Zabbix source**

On your dashboard, create a new Graph panel. Then in the Metrics tab, fill in:

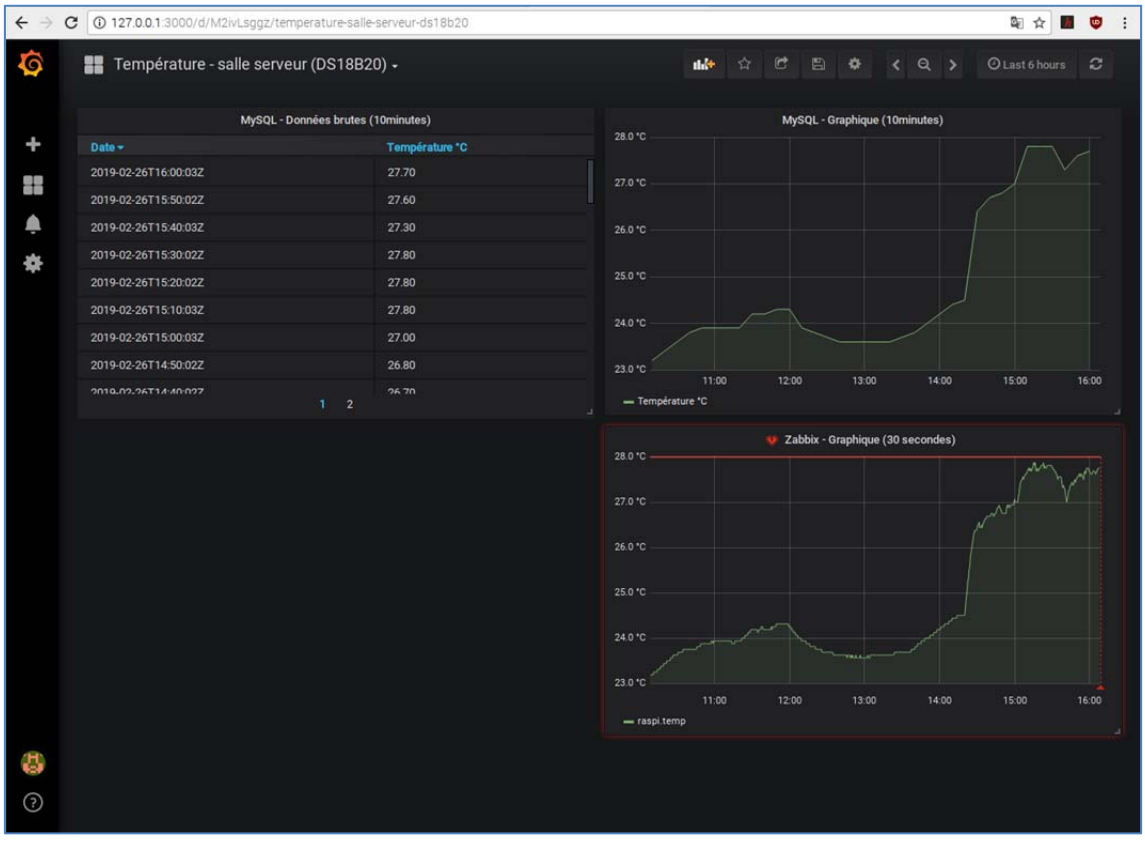
- Data source: Zabbix (here by default)
- Group: Zabbix servers
- Host: Zabbix server
- Item: "raspi.temp" identical to the name of the object or item we had previously created on Zabbix front-end (see capture below), to perform a temperature measurement every 30 seconds.



You should see data from your Zabbix source appear:



It is possible to display on the same dashboard, panels with different sources (here MySQL and Zabbix).



5. Sources

These sources were consulted on February 4, 2019:

- Code4Pi – Les bases : https://code4pi.fr/category/bases/
- Projets DIY – Tutoriels Raspbian : https://projetsdiy.fr/category/mini-pc-ordinateur-carte/raspberry-pi/tutoriels-raspbian-raspberrypi-francais/
- Blog Framboise314 : https://www.framboise314.fr/

These sources were consulted on February 12, 2019:

- Adafruit's Raspberry Pi Lesson 11. DS18B20 Temperature Sensing : https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/software/
- Raspberry Pi Temperature Sensor Web Server : https://wingoodharry.wordpress.com/2014/12/24/raspberry-pi-temperature-sensor-web-server-part-1-intro-sensor-setup-and-python-script/
- Installer un serveur web sur votre Raspberry (Apache + PHP + MySQL) : https://raspbian-france.fr/installer-serveur-web-raspberry-lamp/
- PDO ::__construct : http://php.net/manual/fr/pdo.construct.php/

These sources were consulted on February 20, 2019:

- Download and install Zabbix : https://www.zabbix.com/download?zabbix=4.0&os_distribution=raspbian&os_version=9_stretch&db=mysql/
- Installing Frontend : https://www.zabbix.com/documentation/4.0/manual/installation/install#installing_frontend/
- Zabbix Documentation 4.0 : https://www.zabbix.com/documentation/4.0/manual/
- Stackoverflow - « cannot grant privileges to mysql database » : https://stackoverflow.com/questions/19237475/cannot-grant-privileges-to-mysql-database/
- Raspberry PI – Temperature monitoring with Zabbix : https://sysadmin-ramblings.blogspot.com/2017/04/raspberry-pi-temperature-monitoring.html/
- SQL Data Source Configuration : http://docs.grafana-zabbix.org/installation/configuration-sql/
- Installer Grafana sur Raspbian pour Raspberry Pi : https://projetsdiy.fr/grafana-graphiques-installation-macos-mysensors-influxdb-partie1/
- Using MySQL in Grafana : http://docs.grafana.org/features/datasources/mysql/