# TP3 : Base de données et affichage graphique

Table des matières

[TP3 : Base de données et affichage graphique 1](#_Toc10814327)

[Installation du serveur web : 2](#_Toc10814328)

[Installation D’une base de données (MySQL) : 7](#_Toc10814329)

[Installation de PhpMyAdmin : 8](#_Toc10814330)

[Vérifier l’installation de PHPMyAdmin 9](#_Toc10814331)

[Envoi de données dans notre base de données : [Corrigé complet en fin de tp] 11](#_Toc10814332)

[Une autre base de données, (Influx dB) : 13](#_Toc10814333)

[Installation : 13](#_Toc10814334)

[Rajouter des données dans la base de données influxdb : 15](#_Toc10814335)

[Affichage graphique : 16](#_Toc10814336)

[Grafana : 16](#_Toc10814337)

[Site Web : 22](#_Toc10814338)

[Site WEB : page d’accueil vers toutes les autres visualisations et contrôle de LED : 23](#_Toc10814339)

## Installation du serveur web :

En dernière partie de ce projet, nous allons utiliser notre carte, comme serveur web, et stocker nos données dans des bases de données ( Mysql et Influxdb).

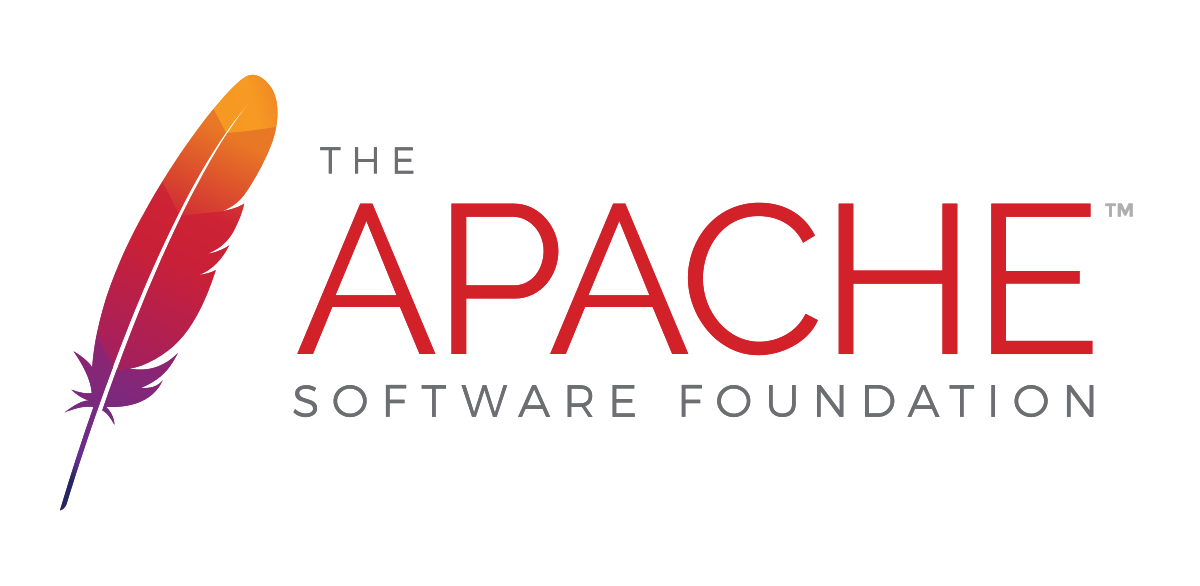
Pourquoi utiliser notre cible au lieu de faire appel à des services qui le font très bien pour nous ?

Plusieurs raisons :

* Economique, en effet pour utiliser notre cible comme serveur web, il suffit juste d’avoir une connexion internet, là ou si on fait appel à un prestataire, il y a des abonnements mensuels/annuels à régler.
* Lorsqu’on utiliser notre propre ordinateur comme serveur web, on a la possibilité de choisir nous-même la taille du disque, l’hébergement de data, etc.

Pour faire un serveur web, et l’utiliser comme base de données on a besoin de plusieurs choses :

* Apache : Apache est un serveur web. En effet, un serveur web s’agit bien d’une machine classique, mais aussi d’un logiciel qui analyse les requêtes des utilisateurs souvent sous forme de requêtes http.



**Installation de Apache :**

**sudo apt install apache2**

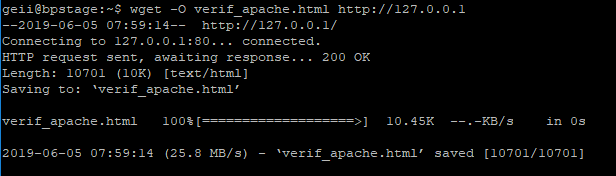
Une fois que ceci est fait, nous allons allouer les droits d’aministration pour le site web à notre utilisateur.

**sudo chown -R geii:www-data /var/www/html/**

**sudo chmod -R 770 /var/www/html/**

Nous allons tenter de nous connecter à notre serveur via le port 80, ce port n’est pas accessible depuis l’extérieur.

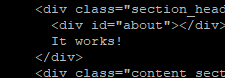
**wget -O verif\_apache.html http://127.0.0.1**



On affiche ensuite le code de la page html :

**cat ./verif\_apache.html**

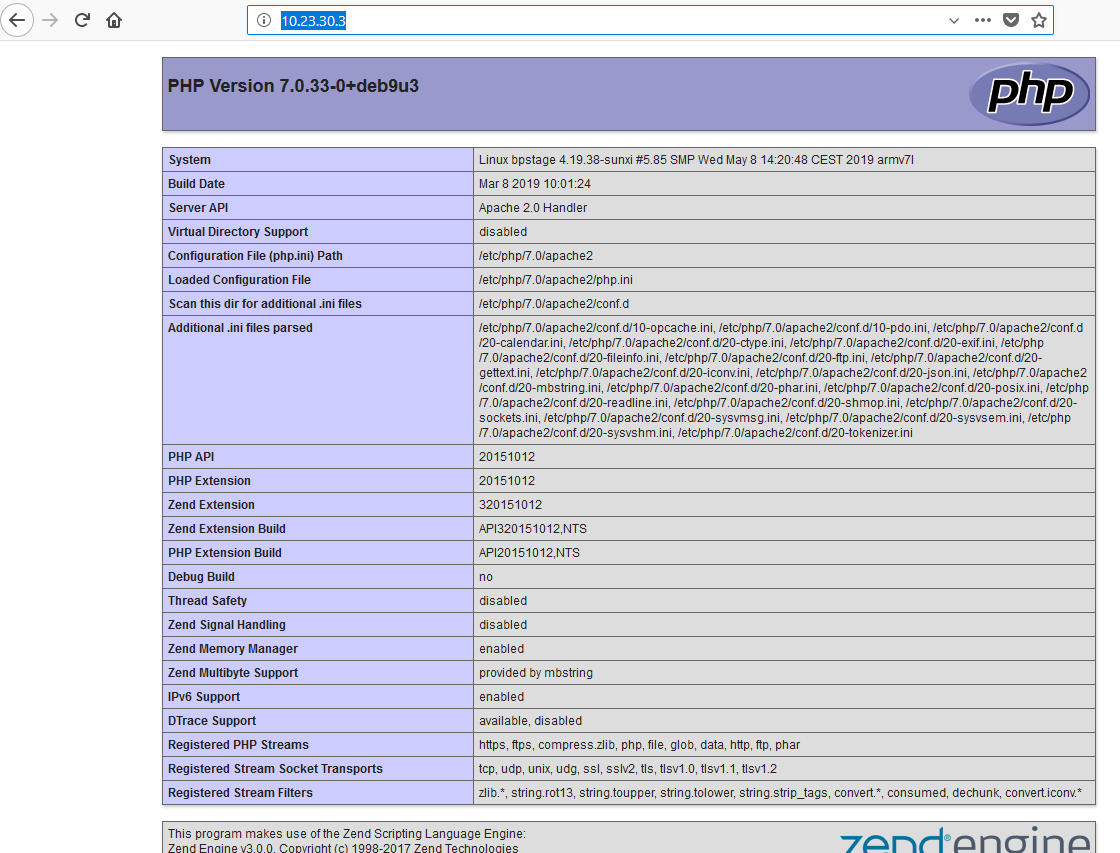
Cherchez dans le code « it works », si vous trouvez, c’est que votre serveur web apache, marche bel et bien.



Vous pouvez aussi faire la verfication directement avec l’interface graphique xfce par exemple à l’aide de vncviewer, (cf tp prise en main).

Ou alors sur un pc sous windows, taper l’adresse ip local de la cible :





Il suffit d’ouvrir Midori ou un autre navigateur web et de taper si vous êtes sous VNC.

<http://127.0.0.1>

It Works devrait alors s’afficher.

Pour ajouter un nouveau fichier, c’est dans le répertoire /var/www/html qu’il faudra le rajouter, car c’est là que cherche apache lorsqu’on appelle le site web.

Nous pouvons déjà dés à présent utiliser notre raspberry pour faire un site en HTML, mais pour compléter notre travail, nous allons rajouter Php.

Tout d’abord, il faut savoir que le **PHP est un langage interprété**. Et comme dans le cas des serveurs, l’acronyme PHP **peut avoir plusieurs sens**. En fait, quand l’on parle de PHP, on peut parler **soit du langage, soit de l’interpréteur.**  
Ici, quand nous parlons d’installer PHP, cela signifie que **nous allons installer l’interpréteur,** afin d’utiliser le langage.

**sudo apt install php php-mbstring**

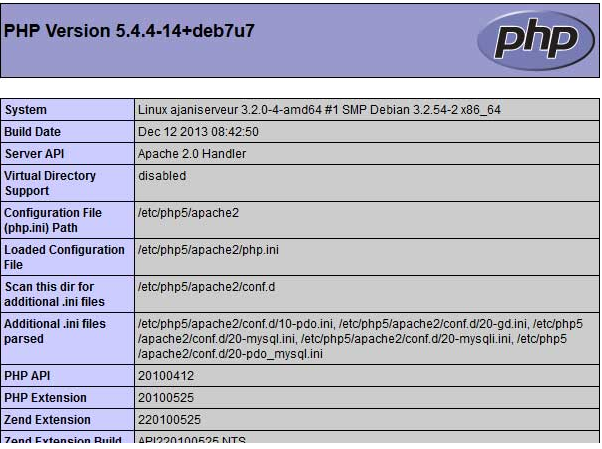
Pour vérifier que php fonctionne, il faut commencer par supprimer le fichier html dans /var/www/html

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

Nous allons maintenant créer le fichier .php

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

Procéder de la même manière que precedemment pour verifier que php fonctionne, chercher si vous restez en ssh « php version », si vous êtes avec vnc, ceci devrait s’afficher :



Nous avons maintenant notre serveur web, ainsi que l’interpréteur PHP.

## Installation D’une base de données (MySQL) :

Nous allons maintenant installer une base de données MySQL à notre serveur afin d’y stocker les données que nous voulons :

MySQL est un système de gestion de base de données (SGBD), il va nous permettre de faire le stockage de toutes informations pour utiliser notre site web.

**sudo apt install mysql-server php-mysql**

Nous allons ensuite nous connecter à notre serveur MySQL :

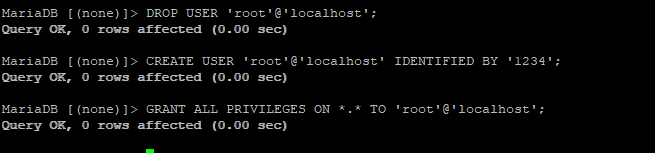
**sudo mysql --user=root**

Une fois connecté, nous allons taper une suite de commande qui va permettre de supprimer **l’utilisateur root et créer un nouvel utilisateur root**, car celui par défaut n’est utilisable que par le compte administrateur du système, et n’est donc pas accessible aux script PHP du serveur.

**DROP USER 'root'@'localhost';**

**CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';**

**GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost';**



Nous avons donc maintenant un **serveur web, relié à PHP et à MySQL**.

(Lors de vos prochaines connexions, vous pourrez donc utilisez la commande

**mysql --user=root --password=votremotdepasse)**

### Installation de PhpMyAdmin :

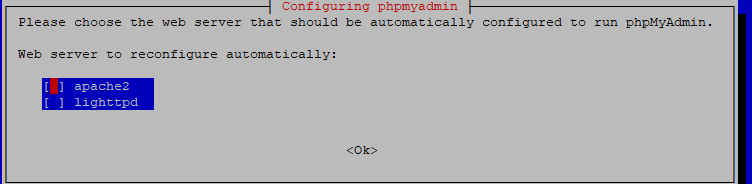
PHPMyAdmin est une application développée en PHP, et qui vise à fournir **une interface simplifiée pour MySQL.**  
Elle vous permet par exemple de voir rapidement et de façon lisible le contenu de votre base de données, ou de la manipuler **sans avoir besoin de faire vous-mêmes vos requêtes MySQL.**

L’installation de PHPMyAdmin se fait très simplement, via le gestionnaire de paquets, en utilisant la commande suivante :

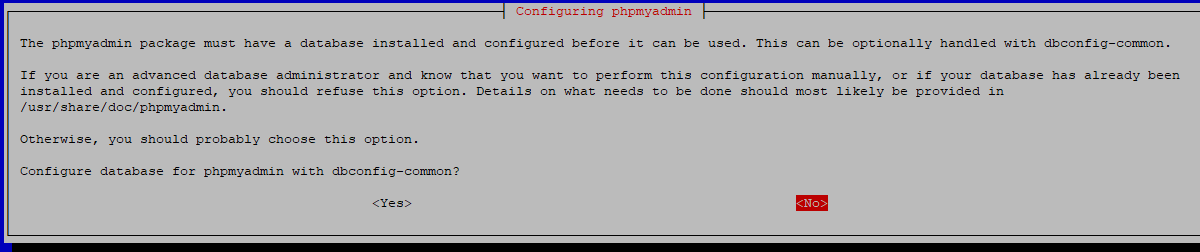
**sudo apt install phpmyadmin**

PHPMyAdmin vous posera plusieurs questions concernant ses réglages.

S’il vous demande de choisir un serveur web, choisir apache2.



Comme nous avons déjà configurez la base de données, choisissez no à la question concernant l’utilisation de dbconfig-common. Choisissez d’utiliser PHPMyAdmin pour un serveur Apache. Pour le mot de passe root, il s’agit de **c**elui que vous aviez utilisé pour MySQL.



Vous devez aussi activer l’extension **mysqli** si cela n’est pas encore fait. Pour cela, utilisez les commandes ci-dessous.

sudo phpenmod mysqli

sudo /etc/init.d/apache2 restart

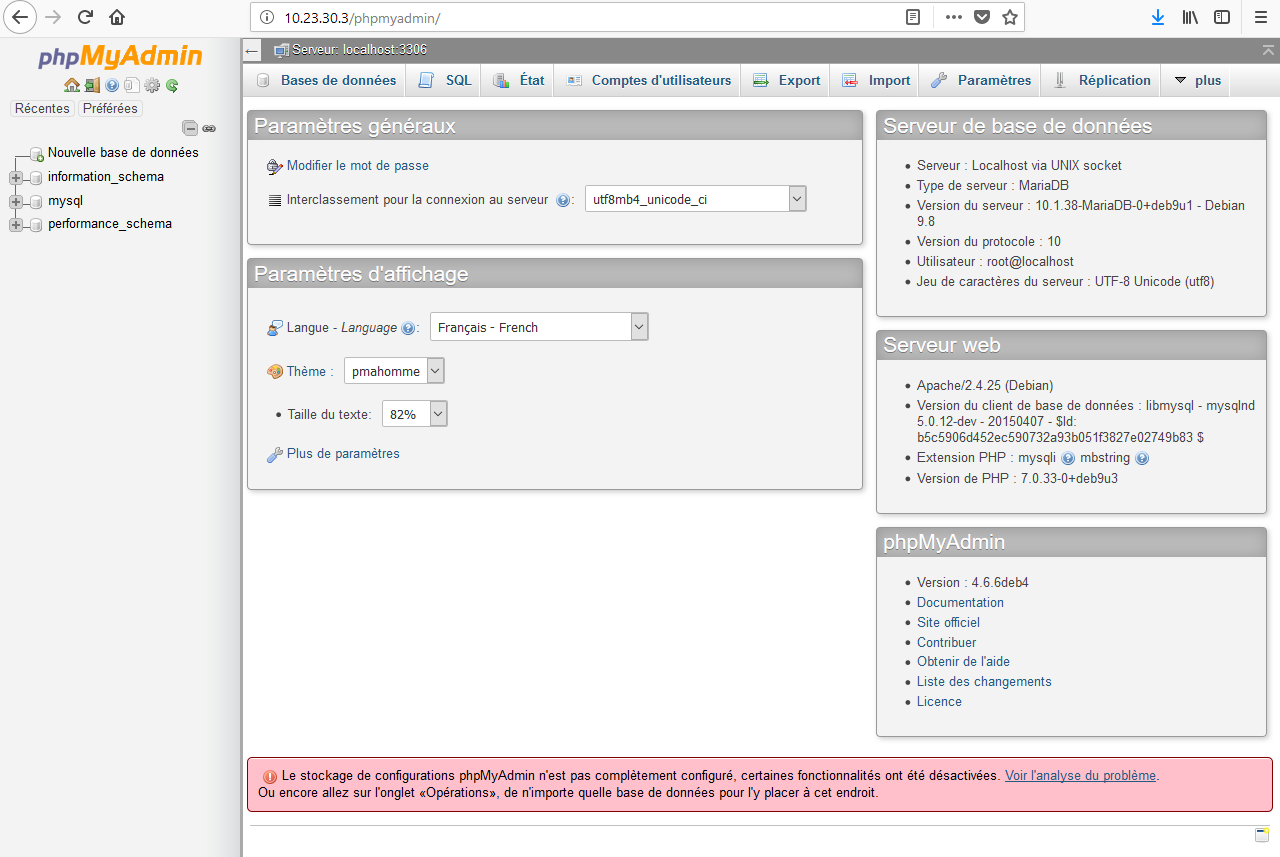
### Vérifier l’installation de PHPMyAdmin

Pour vérifier le bon fonctionnement de PHPMyAdmin, vous allez simple tenter d’y accéder, en utilisant l’adresse de votre Raspberry suivi de /phpmyadmin. Par exemple, en local ce sera « http://10.23.30.3/phpmyadmin ».

Si ça ne marche pas, c’est que phpmyadmin s’est installé dans un mauvais répertoire, il faudra taper la commande suivante :

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

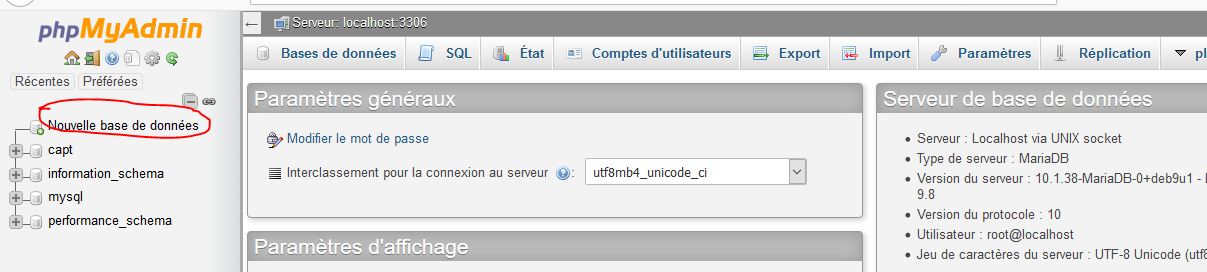
Vous pourrez vous connecter sur votre serveur maintenant avec les identifiants que vous avez tapés précédemment, pour moi c’est : root et 1234



Une fois la base de données installée, nous allons rajouter une fonction à notre programme qui envoie les données de capteur à thingspeak afin d’aussi les enregistrer dans notre base de données sql, dans un premier temps on va utiliser phpmyadmin pour créer une nouvelle base de données.

On pourrait utiliser les fonctions de la librairie sql pour la créer directement avec notre programme, mais l’idée ici est de comprendre qu’un service graphique existe pour faciliter la manipulation des bases de données.

Une fois connecté à Phpmyadmin, créer une nouvelle base de données :



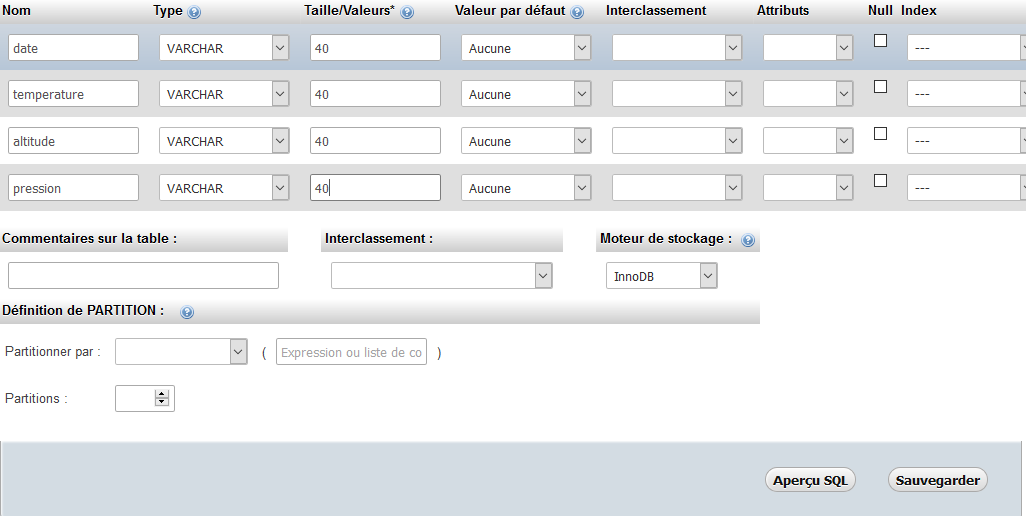
Lui donner un nom, et appuyer sur créer :

Une fois notre base de données créer, nous allons créer une table (tableau), ou l’on va ranger nos données : appelons ce tableau par exemple data :

Nous allons envoyer la data, la température, l’altitude et la pression, choisissez donc 4 colonnes.



Dans notre programme, on range nos données dans des tableaux de caractères, on va les envoyer sous cette même forme, donc dans type choisir VARCHAR avec une taille de 40 pour ne pas avoir de soucis :



Vous pouvez ensuite appuyer sur sauvegarder :

Votre tableau data dans la base de données « nomdevotrebasededonnées » (ici capt) a été créé.

Cependant il est vide, l’objectif maintenant est d’utiliser la libraire sql en C, pour que notre code se connecte à note base de données et rempli notre tableau, juste après avoir envoyé ces mêmes données sur thingspeak.

### Envoi de données dans notre base de données : [Corrigé complet en fin de tp]

. Créez une fonction dans notre code à l’image de la fonction update thingspeak, qui reçoit 3 char.

. Dans cette fonction utiliser ces fonctions pour en premier lieu, vous connecter à votre base de données : Initialisation de la connexion :

MYSQL \*conn;  
 conn = mysql\_init(NULL)

Connexion :

mysql\_real\_connect(conn, "localhost", "root", "1234", "nomdb", 0, NULL, 0)

Création d’un tableau de caractère ou l’on va mettre toute la syntaxe de l’appel de la fonction d’envoi :

char query[1000];

snprintf(query, 1000, "INSERT INTO data(date,temperature, altitude,pressure) VALUES(NOW(),%s,%s,%s)",v, v2, v3);

La fonction NOW() renvoie la date et l’heure actuelle du système, si l’heure renvoyée n’est pas la bonne :

Taper sur votre cible :

**sudo timedatectl set-timezone Europe/Paris**

Envoi du tableau :

mysql\_query(conn, query);

Femeture de la connexion :

mysql\_close(conn);

Notez bien : il faut utiliser la libraire sql :

Header : #include <mysql/mysql.h>

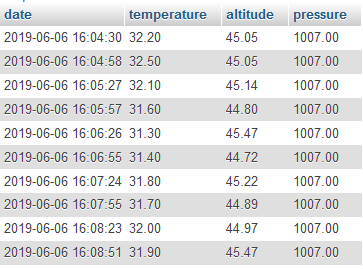
Suffixe de compilation : -lmysqlclient :



Se référer à la fonction à l’appel de la fonction dans le main : Updatesql pour correction :

<https://gist.github.com/Mizaystom/9ac852ce8b30d8419e81834029dec2d>

Une fois que vous avez reçu, à chaque fois que vous recevrez une donnée de l’arduino, la led s’allumera en vert, et le banana pi va envoyer à thingspeak, mais aussi remplir la base de données.



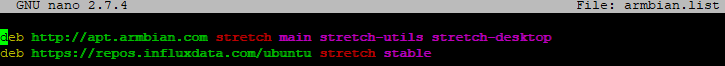
## Une autre base de données, (Influx dB) :

### Installation :

Tout comme sql, Influx dB est un serveur de base de données.  
La différence réside dans le fait que influx dB est désigné pour gérer des données temporels.  
En effet même si sql peut aussi gérer les données avec le temps, son objectif principal est simplement de générer une base de données brute.  
Utiliser influx dB, revient donc à utiliser sql en mettant comme première colonne de la table « date ».

Pour télécharger cette nouvelle base de données, il faut d’abord rajouter le « repository » à la liste des répertoires d’installation en allant dans /etc/apt/sources.list.d/armbian.list et en copiant :

deb https://repos.influxdata.com/ubuntu stretch stable



Ensuite, sachant que ce répertoire n’est pas libre d’accés, comme le principal, il faut rajouter sa clé :

**curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add –**

Une fois le répertoire, et son mot de passe importés on peut installer influxdb :

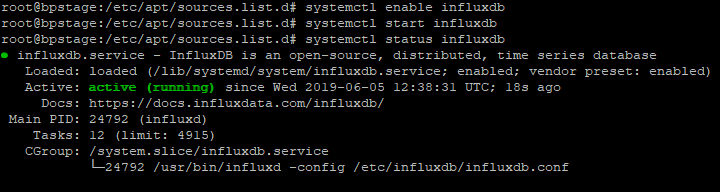
**apt update && sudo apt install influxdb**

Puis on active le service pour le lancer automatiquement aux prochains démarrages et on lance le service.

**enable influx db**

**start influx db**

**status influxdb**

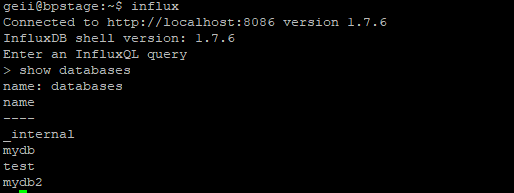


Une fois votre service lancé, vous pouvez créer une base de données avec une requête poste :

curl -XPOST "http://localhost:8086/query" --data-urlencode "q=CREATE DATABASE mydb"



Pour vérifier qu’elle existe, on peut lancer le service influx et ensuite taper la commande show databases :



On peut ensuite créer la base de données en lançant le service, et invoquant la méthode CREATE sans passer par la requête X POST.

### Rajouter des données dans la base de données influxdb :

Le site web de influx db, nous renseigne sur la méthode pour stocker des données dans notre base de données :

<https://docs.influxdata.com/influxdb/v1.7/guides/writing_data/>

curl -i -XPOST 'http://localhost:8086/write?db=mydb' --data-binary 'cpu\_load\_short,host=server01,region=us-west value=0.64 1434055562000000000'

Comment on peut le voir sur l’exemple, on utilise une nouvelle fois la méthode XPOST.

On renseigne l’url localhost :8086 et on indique dans qu’elle base de données on souhait écrire db=xxxxx ensuite après –data-binary, il y a plusieurs éléments entre les guillemets,

Le premier cpu-load\_short, est le nom de la donnée que l’on veut envoyer, ensuite il y a des tags, host et region, qui ne sont pas obligatoires, et finalement il y a la valeur et en dernier le « timestamp », c’est le temps en nanosecondes depuis le 1er Janvier 1970 ? UTC, il est aussi optionnel.

Exercice : créer une data base test, et envoyé une valeur : <https://docs.influxdata.com>

Une fois que cet exercice est fait, tout comme la fonction fait pour mysql, rajouter une fonction dans notre programme en utilisant curl pour envoyer les données du capteur à la base de données influxdb, la correction est dans la fonction update influx.

<https://gist.github.com/Mizaystom/9ac852ce8b30d8419e81834029dec2d1>

## Affichage graphique :

### Grafana :

Nous allons voir deux manières différentes d’afficher les données stockées dans nos bases de données.

Premièrement, nous allons utiliser grafana :



Grafana est un parmi plusieurs logiciels dit de supervisassions.

**Grafana** est un logiciel libre qui permet la visualisation et la mise en forme de données métriques. Il permet de réaliser des [tableaux de bord](https://fr.wikipedia.org/wiki/Dashboard) et des graphiques depuis plusieurs sources dont des [bases de données de série temporelle](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_temporelle) ([Time Series Database](https://fr.wikipedia.org/w/index.php?title=Time_Series_Database&action=edit&redlink=1)) comme Graphite, [**InfluxDB**](https://fr.wikipedia.org/wiki/InfluxDB)et [OpenTSDB](https://fr.wikipedia.org/wiki/OpenTSDB).

Pour installer grafana, il faut une nouvelle fois rajouter le repository dans la liste des répertoire ou va chercher la cible lors d’une installation.

Ajout dans le repository de l’adresse de la gestion de paquet de grafana

**echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee /etc/apt/sources.list.d/grafana.list**

ajout de la clé de validation

**curl https://packages.grafana.com/gpg.key | sudo apt-key add -**

**sudo apt update && sudo apt-get install -y apt-transport-https grafana**

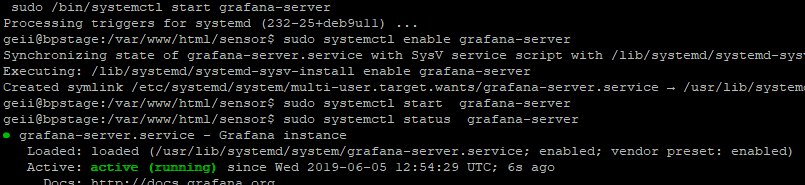
On active le lancement automatique de grafana à chaque démarrage :

**sudo systemctl enable grafana-server**

On lance et on montre le status pour vérifier :

**sudo systemctl start grafana-server**

**sudo systemctl status grafana-server**



Vérifions si ça fonctionne

<http://10.23.30.3:3000>

Login et passwd = admin

Puis on change le passwd : apprentis

Une fois lancé, faire :



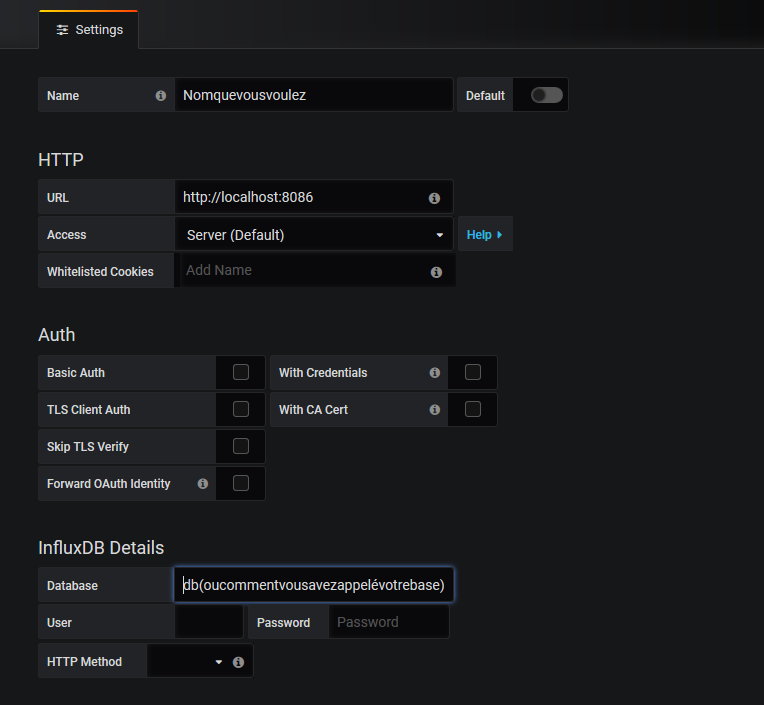
Puis choisir « Data Sources »



- Cliquer sur add data source



- Choisir InfluxDB



Remplir comme ci-dessus.

Ensuite, nous allons accéder au dashboard :



Choisir la visualisation que vous préférez :



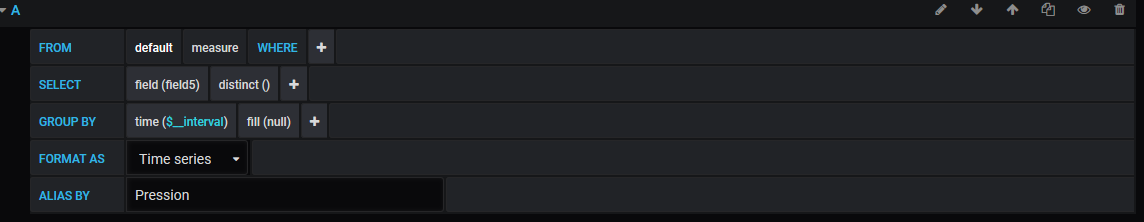
Ensuite en fonction de votre requête xpost, remplir les query :

Sur from : choisir le nom que vous avez donné (dans le code measure) :

Dans select, choisir la donnée à afficher sur le graph, par exemple dans field5, il ya la pression :

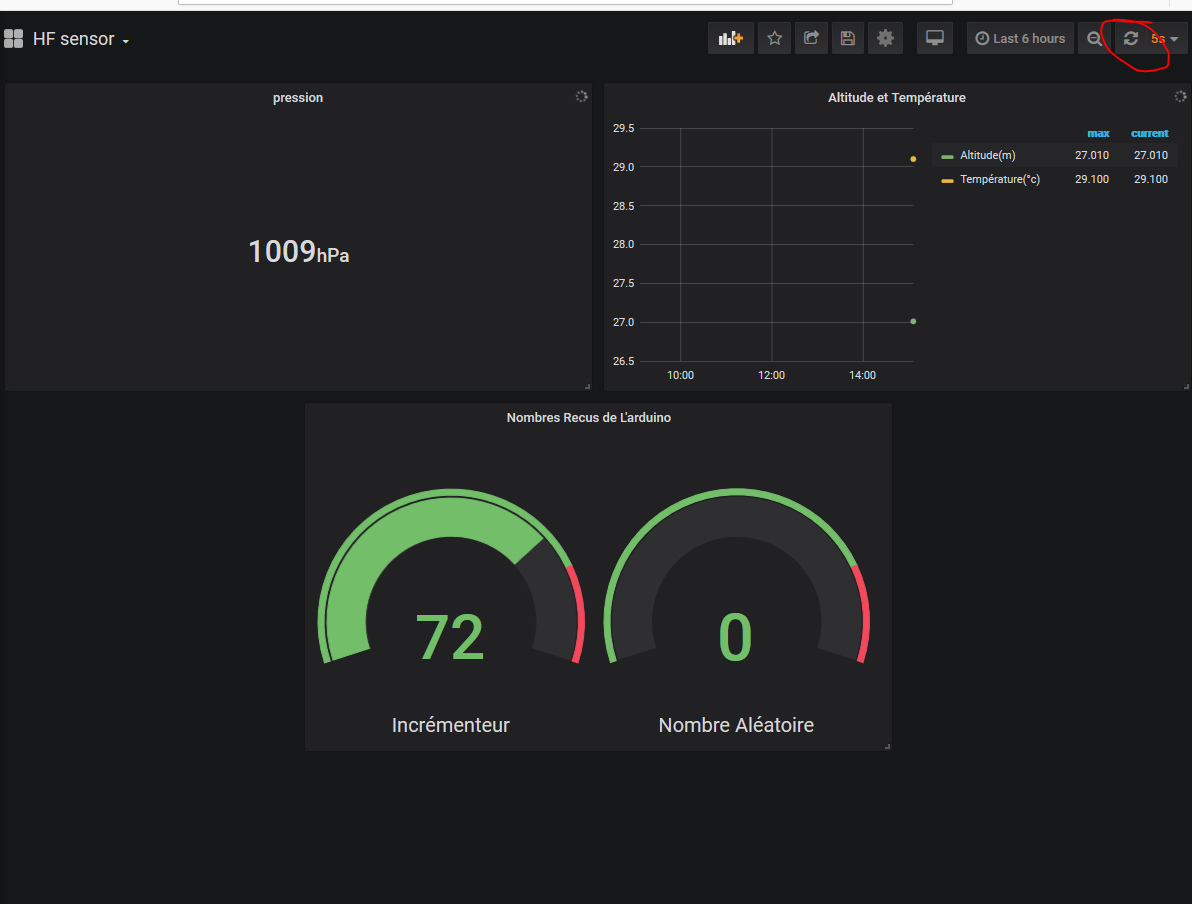
Choisir distinct() ou mean() selon l’affichage que vous préférez.

Dans Alias mettre le nom de la légende :

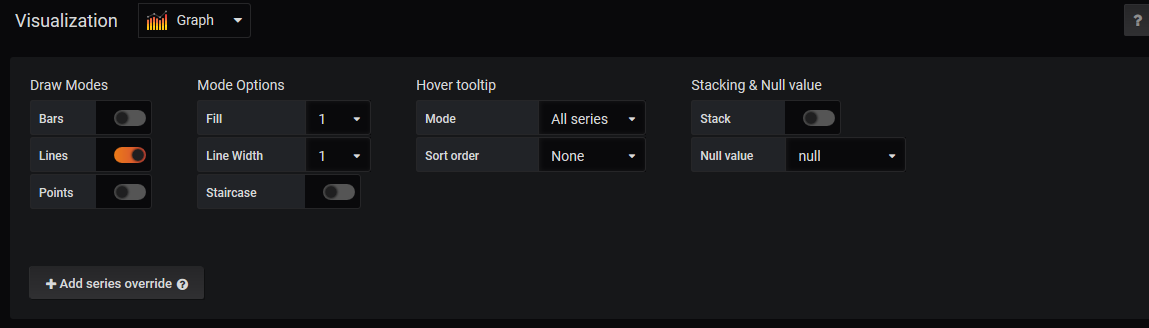


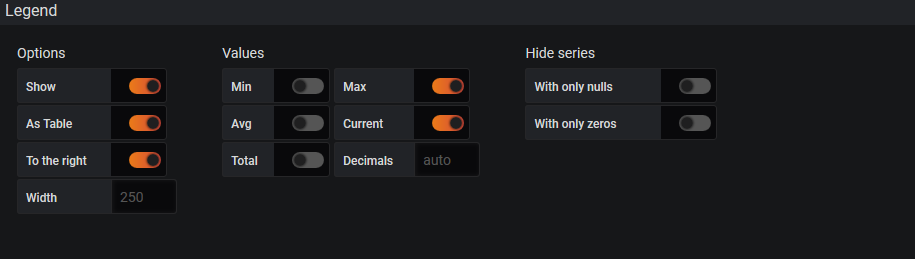
Faire ceci, pour tous les Fields et lancer le programme :

En haut à droite, choisir le temps de rafraîchissement :



Vous pouvez adapter vos visualisations dans l’onglet visualisations :





### Site Web :

Nous avons vu l’affichage à l’aide d’un serveur fait pour cela, maintenant nous allons nous intéresser à faire une page web en prenant cette fois comme exemple la base de données MySQL.

On va pour cela supprimer l’index.php dans /var/www/html, et en créer un nouveau.

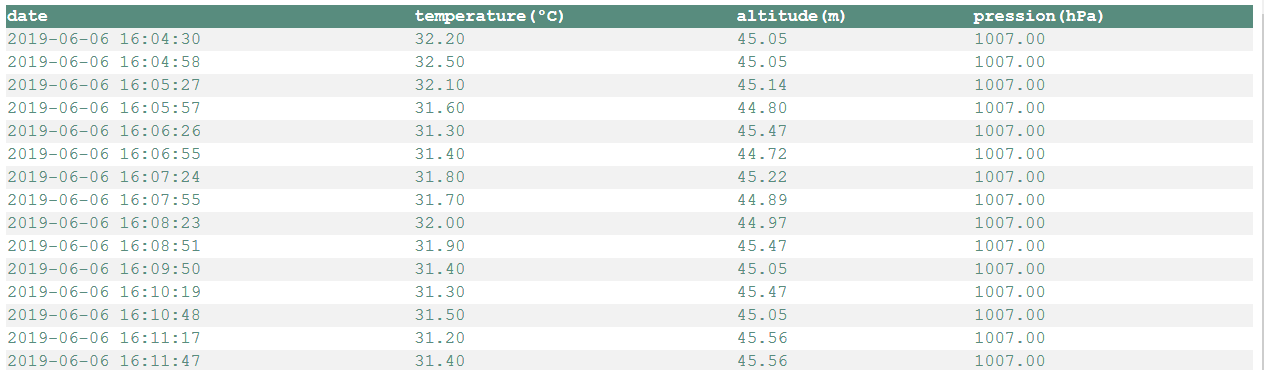
Adapter ce code à notre base de données sql :

<https://www.codeandcourse.com/2017/11/how-to-display-data-from-mysql-database-into-html-table-using-php/>

Correction :

<https://gist.githubusercontent.com/Mizaystom/fa5a622616f789e24163dc039bd92ed4/raw/87b66c818a269b023d02510b97ba4d8e92e2ad5d/index2.php>

Résultats attendus :



Pour y accéder : 10.23.30.3

### Site WEB : page d’accueil vers toutes les autres visualisations et contrôle de LED :

. Créer un dossier dans /var/www/html et déplacer notre index.php crée précédemment dedans de sorte à ce qu’on y accéde en tapant par exemple 10.23.30.3/sensor

.dans /var/www/html, créer un nouvel index.php

Nb : il est pertinent d’appeler nos fichiers index.php car lorsque apache cherche un fichier à afficher, il cherche d’abord : index.html puis index.php.

Le fait d’appeler nos fichiers index lui permet d’affichier la page directement, car la norme est d’appeler la page d’affichage index

Copier ce code : <https://gist.githubusercontent.com/Mizaystom/fa5a622616f789e24163dc039bd92ed4/raw/87b66c818a269b023d02510b97ba4d8e92e2ad5d/index.php>

Dans notre page index.php

Expliquons le code :

<input type="submit" style = "font-size: 16 pt" value="OFF LEDS" name="offg">

On crée un bouton, de type envoie, qui va afficher OFF LEDS et dont le nom est offg.

<?php

Balise php, le langage php nous permet de controller des page webs non statiques, il va nous aider à utiliser une commande sur la cible ou se situe le serveur :

shell\_exec("gpio mode 7 out ");

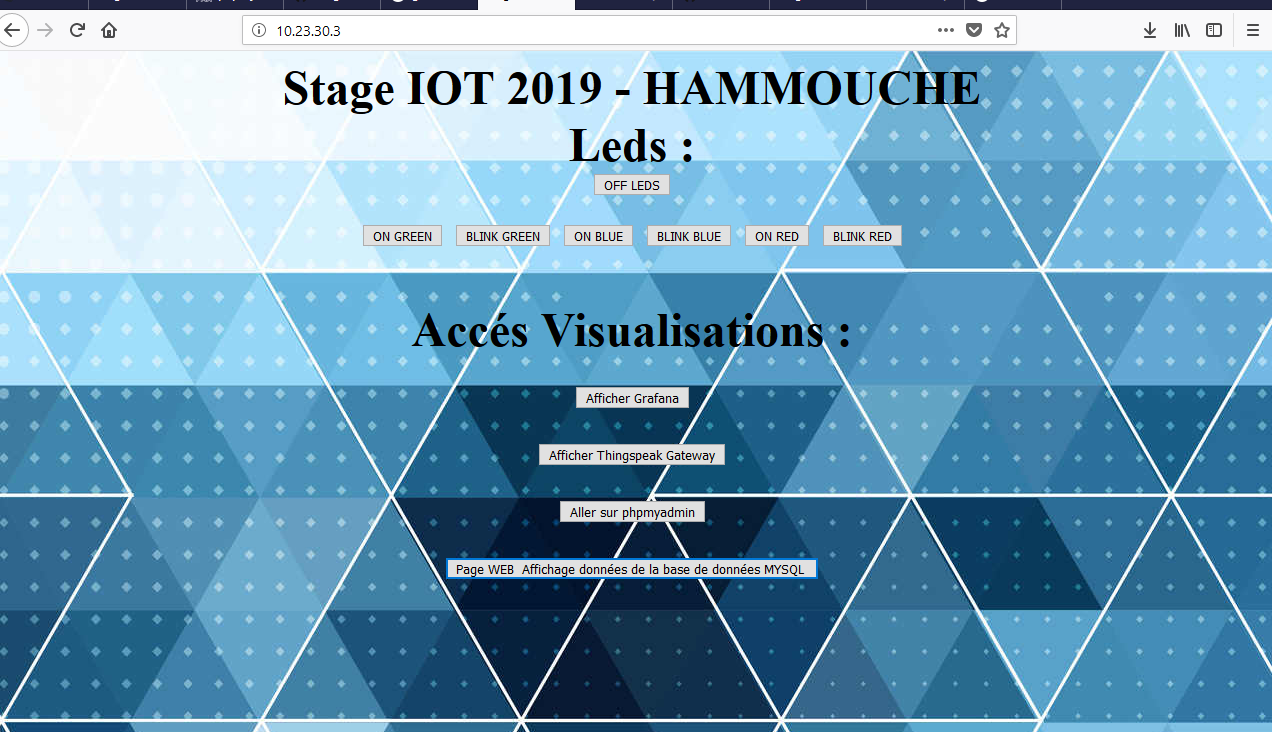
La commande shell\_exec, permet d’envoyer sur notre cible la commande taper dans les guillemets,

Dans notre cas, on met la gpio 7 en sortie, cette commande necessite la bibliothéque wiring pi dans notre cible, qui a été installé lors d’un precedent TP.

<a href="http://10.23.30.3/phpmyadmin" target="\_blank"> <input type="button" value="Aller sur phpmyadmin"> </a>

On crée un bouton qui renvoie à l’adresse indiquer après href =, ce bouton s’appelle phpmyadmin.

Au final, nous avons une page web comme ceci :



On peut controller les leds, et acceder à nos affichages grafana, thingspeak et web à partir de cette page principale