Environments Monitoring & Management Application

EMMA is a web application developed by Antonio Díaz Pozuelo (adpozuelo@gmail.com) whose objective is to monitor and manage Technical Scientific Facilities (ICT).

Design and architecture

EMMA has been designed and implemented using two main components:

1. Sensors (Python): “scripts” tailor-made implemented, which are responsible of collecting the sensor metrics and saving them in the database. These programs are installed on a Raspberry Pi, which has the corresponding sensor connected, or on a computer or node to be monitored. Sensors are currently implemented for DHT11/DHT22 (temperature and humidity), KY038 (sound), computer or computing node (with and without GPU).
2. LAMP Server (Linux+Apache2+MariaDB+PHP):
    1. Web Application (CodeIgniter+Bootstrap+ChartJS): allows us to monitor and manage ICT through a web interface that adapts to all browsers and electronic devices (responsive web).
    2. Database (MariaDB): which stores the metrics collected by the sensors and all the data necessary for the proper functioning of the web application.
    3. Management of alerts and reports (Python+Telegram): "scripts" that send alerts to mobile phones (via Telegram) in the event of sensor failures (device or sensor crash) and/or metrics (high temperature or humidity, sound from an alarm, etc.). In addition, every certain interval, the server sends a statistical report of the desired metrics to the mobile phones that belong to the indicated Telegram group.

Installation

On the one hand, the installation of the server is done as follows (Ubuntu 20.04):

1. Installing Python >= 3.6:
   ```
   $ apt-get install python3
   ```
2. Installation of required packages:
   ```
   $ pip install psutil gputil py-cpuinfo peewee PyMySQL
   ```
3. Installation and configuration of the database:
   ```
   $ apt-get install mariadb-server
   $ mysql_secure_installation
   $ mysql -u root -p
   > CREATE DATABASE emma;
   > CREATE USER 'emma'@'%' IDENTIFIED BY 'yourpassword';
   > CREATE USER 'emma'@'localhost' IDENTIFIED BY 'yourpassword';
   > GRANT ALL PRIVILEGES ON emma.* to 'emma'@'%';
   > GRANT ALL PRIVILEGES ON emma.* to 'emma'@'localhost';
   > SELECT user,host FROM mysql.user;
   > SHOW GRANTS for emma;
   ```
4. Installation and configuration of Apache2:
   ```
   $ apt-get install apache2 php php-mysql libapache2-mod-php \
       php-json php-mbstring php-mysqlnd php-xml php-intl php-curl
   $ a2enmod rewrite
   $ nano /etc/apache/sites_enabled/000-default
   + <Directory "/var/www/html/public">
   ```

```
+          Options Indexes FollowSymLinks
+          AllowOverride All
+         Require all granted
+ </Directory>
```

5. Unzip CodeIgniter 4 (https://codeigniter.com/) in the Apache2 web directory:
   ```
   $ unzip codeigniter4-framework-v4.3.1-0-gb1e5c64.zip
   $ mv codeigniter4-framework-b1e5c64/* /var/www/html/.
   ```
6. Unzip the EMMA application inside the CodeIgniter framework:
   ```
   $ cd /var/www/html
   $ tar zxvf ~/EMMA.tgz
   $ chown -R www-data:www-data *
   $ systemctl restart apache2.service
   ```
7. Configure EMMA:
   ```
   $ nano app/Database/Seeds/Init.php ← (Set your admin credentials).
   $ nano env ← (Configure your database configuration).
   $ cp env .env
   ```
8. Copy the following “scripts” in /usr/local/bin and add the following lines in CRON:
   ```
   $ cp  sensors/check_alarms.py /usr/local/bin/.
   $ nano /usr/local/bin/check_alarms.py ← (Change GROUP to your Telegram group).
   $ cp  sensors/check_cluster.py /usr/local/bin/.
   $ cp  sensors/sensors_report.py /usr/local/bin/.
   $ nano /usr/local/bin/sensors_report.py ← (Change GROUP to your Telegram group).
   $ cp sensors/emma_model.py /usr/local/bin/.
   $ nano /usr/local/bin/emma_model.py ← (Configure your database configuration).
   $ export EDITOR=nano && crontab -e
   + SHELL=/bin/bash
   + */1 * * * * /usr/local/bin/check_alarms.py&>>/var/log/emma.log
   + */10 * * * * /usr/local/bin/check_cluster.py&>>/var/log/emma.log
   + 0 20 * * * /usr/local/bin/sensors_report.py&>>/var/log/emma.log
   ```
9. Rotation of logs:
   ```
   $ nano /etc/logrotate.d/emma
   + /var/log/emma.log {
   + rotate 4
   + weekly
   + compress
   + missingok
   + notifempty
   + }
   ```
10. Initialize EMMA to load the administrator credentials:
    ```
    $ cd /var/www/html
    $ php spark db:seed Init
    ```
11. You can modify the HOME of the web in the following file app/Views/home.php
12. EMMA is installed and configured to access via http://localhost

On the other hand, the installation of clients (sensors or equipment) is done as follows (Ubuntu 20.04 or Raspberry Pi OS):

1. Installing Python >= 3.6:
   ```
   $ apt-get install python3-dev python3-pip
   ```
2. Installation of required packages:
   ```
   $ python3 -m pip install --upgrade pip setuptools wheel
   $ pip3 install adafruit-circuitpython-dht psutil peewee PyMySQL ← (Raspberry Pi OS)
   $ pip3 install psutil gputil py-cpuinfo peewee PyMySQL ← (Ubuntu 20.04)
   ```

3. On Raspberry it is recommended to disable Wifi/Bluethoot (if it is not going to be used) and remove Avahi-Daemon:
   $ apt-get remove avahi-daemon
   $ nano /etc/sysctl.d/99-sysctl.conf
   + net.ipv6.conf.all.disable_ipv6 = 1
   + net.ipv6.conf.default.disable_ipv6 = 1
   + net.ipv6.conf.lo.disable_ipv6 = 1
   $ sysctl -p
4. Copy the desired sensor from the “sensors” directory to the $HOME directory (Raspberry Pi OS):
   $ cp sensors/emma_model.py ~/.
   $ nano ~/emma_model.py ← (Configure your database configuration).
   $ cp sensors/dht11_sensor.py ~/. ← DHT11 sensor.
   $ cp sensors/dht22_sensor.py ~/. ← DHT22 sensor.
   $ cp sensors/KY038_sound_sensor.py ~/. ← KY038 sensor.
   $ cp sensors/node_sensor.py ~/. ← Sensor equipment or node.
5. Configure CRON (example with DHT11 sensor to record readings every 5 minutes):
   $ export EDITOR=nano && crontab -e
   + SHELL=/bin/bash
   + */5 * * * * ~/dht11_sensor.py&>>/home/pi/dht11_sensor.log
6. Log rotation:
   $ nano /etc/logrotate.d/dht11_sensor
   + /home/pi/dht11_sensor.log {
   + rotate 4
   + weekly
   + compress
   + missingok
   + notifempty
   + }

For any questions or suggestions, please send an email to adpozuelo@gmail.com.

Symbology:

$ → Command to execute from the command interpreter of the operating system.
> → Command to execute from the MariaDB command interpreter.
+ → Line to add in the edited file.