SHELLY POWER OUTAGE MONITORING

Scenario:

During a power outage in a remote location, the user won’t receive any notification about the situation, and the time to take action can be longer than needed.

Solution:

With 2 plugs, one plug (a) connected to the grid and one plug (b) connected to a UPS station together with a router connected to the internet, we guarantee that power will be always on in case of an outage. The plug “b” will test in a regular interval set by the user if plug “a” is active.

In case of plug “a” can’t communicate with plug “b”, this will send an email to the user and with that information, the user can actively take actions to confirm the outage and dispatch a team to monitor the situation instead of not knowing what is happening at that remote location.

With that, we want to improve the response time by allowing the user to become an active actor in the situation.

We provide two solutions:

1. Run the script responsible for sending the emails locally, in a machine connected to the outlet that will be monitoring thus being always powered.
2. Run the script on a server outside the location.

For either solution, an internet connection is mandatory and the computer that will run the server must have Node JS installed.

To run the email server script, follow the steps below:

1. Unzip the shelly\_Outage\_Monitoring.zip on your computer.
2. Open the folder shelly\_email\_server
3. Open the command prompt in the folder location
4. Execute the command “node index.js”
5. The email server is running and ready to send/receive emails.

To setup the email server script to use your credentials:

1. Inside the folder shelly\_email\_server, open the file secrets.js
2. Change the service to your own.

const hotmail = {

  EMAIL\_SERVICE: 'hotmail',

  EMAIL\_USERNAME: 'your\_handler@outlook.com',

  EMAIL\_PASSWORD: 'your\_password',

  EMAIL\_FROM: 'your\_handler@outlook.com',

}

1. For more information about how to configure the service or set the message, visit <https://nodemailer.com/>
2. This file stays on the server and is not accessible without the proper authorization to login into the server.
3. Google offers a specific credential to be used only by the application and that has to be set on Google first to get the correct credentials.

To run the script in a shelly device:

1. Copy the contents of the file shelly\_script.js
2. In your Shelly device, click on “< > Scripts”.

Graphical user interface, application

Description automatically generated

1. Click on “Add script”

Icon

Description automatically generated

1. It will open the coding interface. Give it a name and paste the content in the coding area.

Graphical user interface

Description automatically generated with medium confidence

1. It should look like this

Graphical user interface, text, application

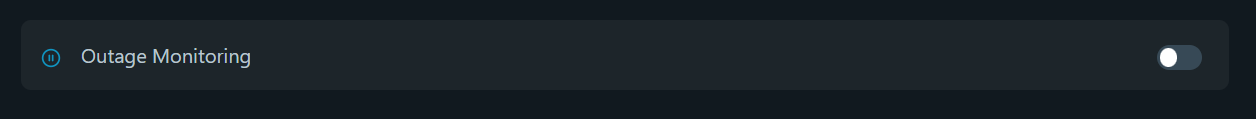
Description automatically generated

1. Make the necessary modifications (check SCRIPT SETUP topic below)
2. Click on “Save”
3. Click on “<> Scripts” to go back to the script list

Background pattern

Description automatically generated

1. Click on the play icon to start



These steps are standard for any script that is developed outside the device interface.

SCRIPT SETUP

To set up the script we need to modify the variables in the code.

1. Email – Replace the current address with the one that will receive the notifications.
2. targetIp – Add as many valid device IP addresses as you want. The format must be ['192.168.15.35', '192.168.15.35', '192.168.15.35', '192.168.15.35']. the IP must be between quotes and divided by a comma all inside [ ].
3. loopTime – Time in seconds. Use multiplication to scale into hours, i.e. 60\*60\*12 translate to 12 hours. The device will run the script every 12h.
4. opt – Two options:
   1. False – This means it will only send emails when there is an outage.
   2. True – This means it will send an email even when the power is on.
5. computerIp – Two options:
   1. Run in a remote server. For that to happen, replace the IP with the IP of your server. You need to open port 60222 in your firewall and redirect traffic in your network to port 80.
   2. Run inside your local network. Replace the IP with the IP of the machine running the server. Unless required by the infrastructure, there is no need to open ports in the firewall.

This is what the script variables look like in the code:

// EDIT THESE PARAMETERS

        // email of the person receiving the notification

        let email      = 'demonstration@allterco.com';

        // IP list of all devices that will be monitored

        let targetIp   = ['192.168.15.35'];

        // number in seconds that will take to check each device in the list

        let loopTime   = 60\*60\*12;

        // send emails even when the device is connected

        let opt = false;

        // email server configuration. Two options. Comment the line that won't be used

        // 1) ip address of the computer that will run the email server locally

        // let computerIp = '192.168.15.99';

        // 2) Shelly public ip address of our server that is running the email server via internet

        let computerIp = '23.115.145.128';

Once these are modified, proceed to run the script in the device.