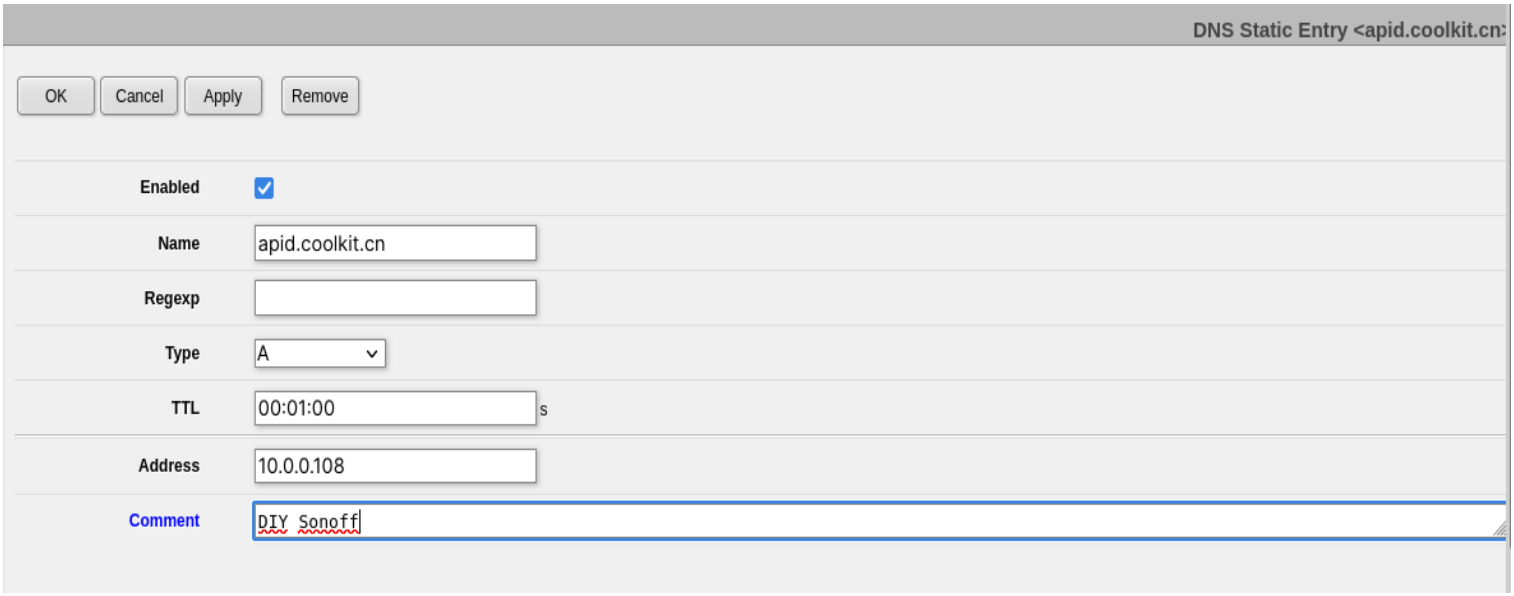
The procedure consists in sending commands to a server on the LAN emulating the non-functioning online one.

My router is 10.0.0.1
my apache server is 10.0.0.108
my sonoff is 10.0.0.118

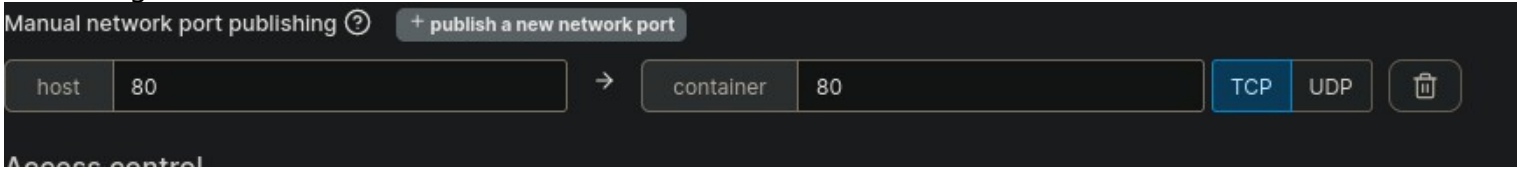
Spoofing the apid.coolkit.cn domain (it can change based on the original firmware loaded on the sonoff, always check it by running tcdump on the packets) using a mikrotik router this is how it is done

		#	Name	Regexp	Type	TTL (s)
- D		0	apid.coolkit.cn		A	00:01:00

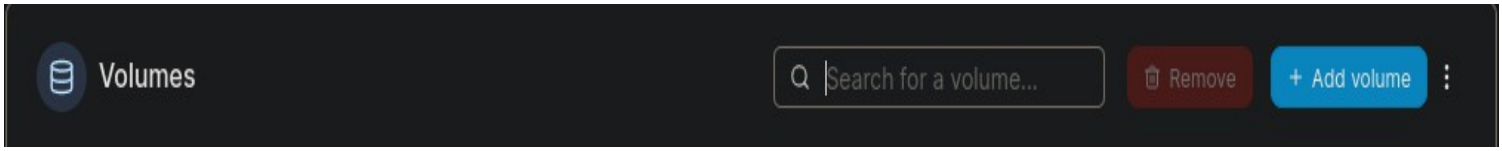


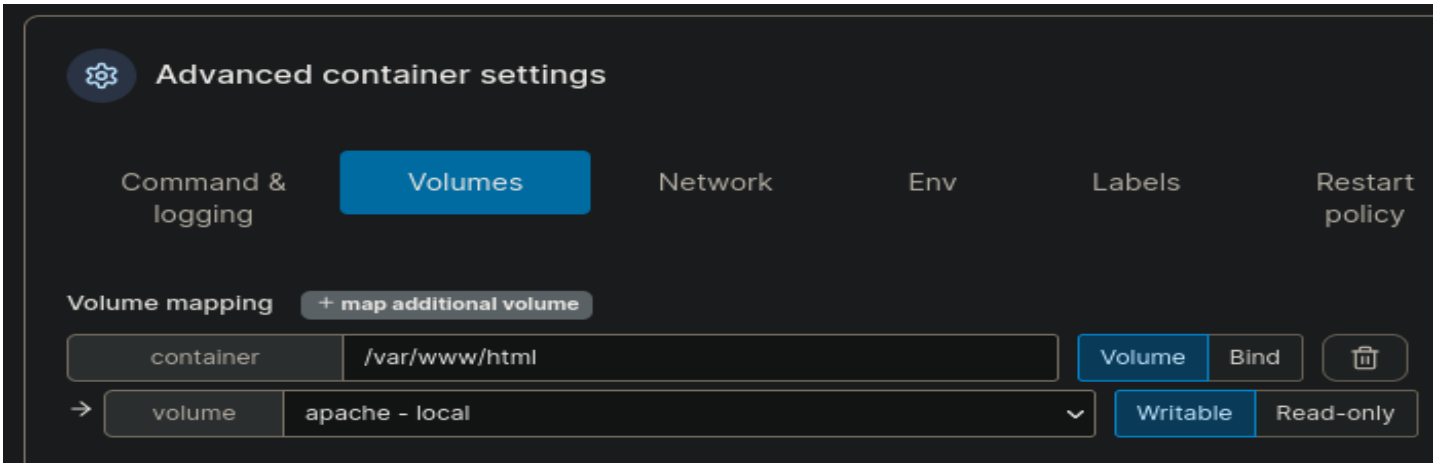
now we need to create the apache server with integrated php using docker and with the portainer graphical interface it is very simple:
let's create a container

let's configure the door

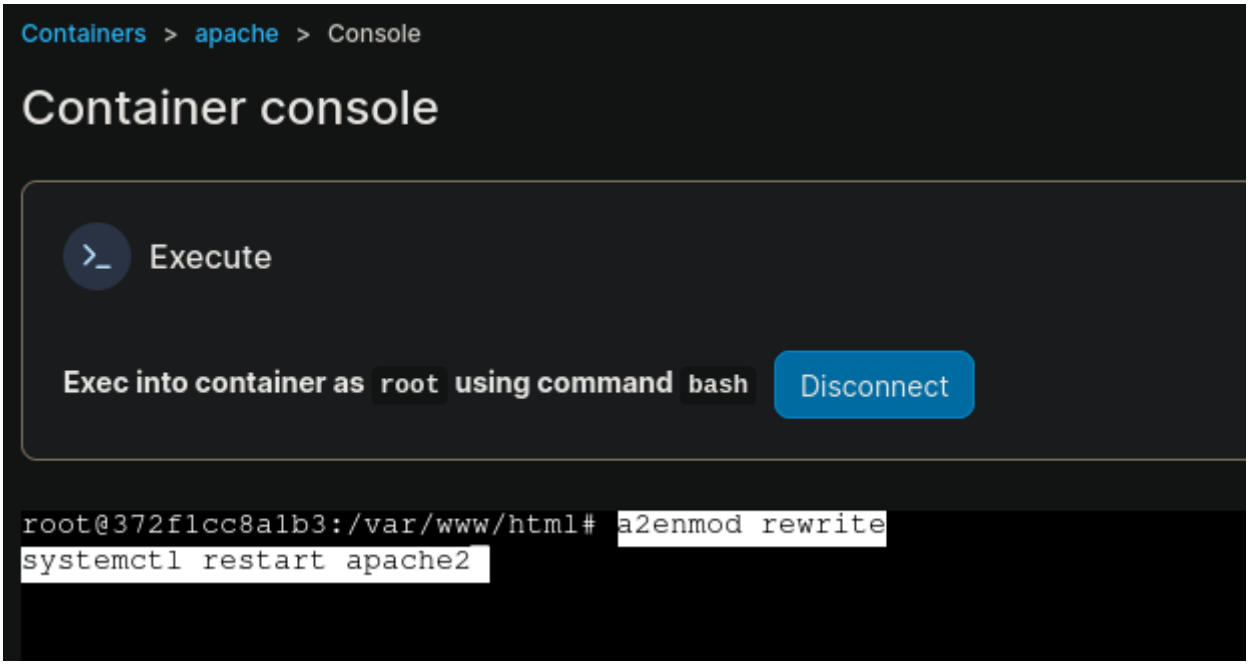


then the volume where the content of our emulated site will be

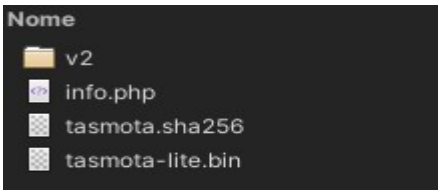


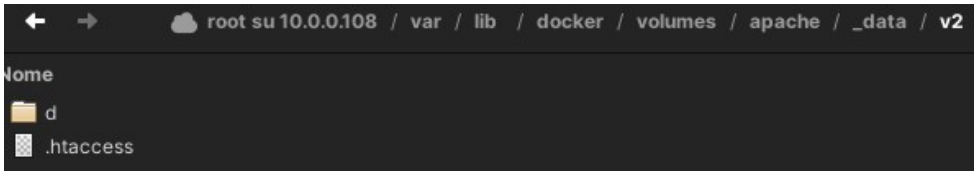


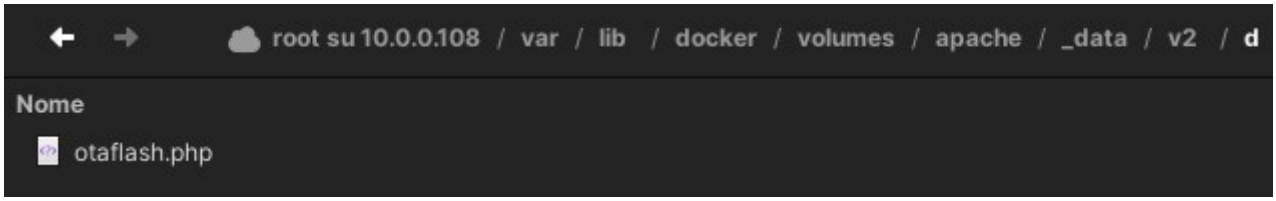
we enable rewrite (to read the .htaccess files) and restart Apache by passing the commands below individually in the portainer console



now we copy, again from the console, or download with wget the tasmotalite software and its sha256 into / var/ww/html
then we create the subdirectories v2--->
d and inside v2 we put .htaccess and
inside d the otaflash.php file







this is the contents of .htaccess

```
# Turn on the rewrite engine
RewriteEngine on
RewriteBase /
# If the request doesn't end in .php (Case insensitive) continue processing rules
RewriteCond %{REQUEST_URI} !\.php$ [NC]
# If the request doesn't end in a slash continue processing the rules
RewriteCond %{REQUEST_URI} [^/]$
# Rewrite the request with a .php extension. L means this is the 'Last' rule
RewriteRule (.*)$ $1.php [L]
```

this is the content of otaflash.php

```
<?php
echo '{
   "error": 422
}';
?>
```

these files must be written with a pure editor like geany in linux or nano or vi with the command :set nobomb :wq

---

Now let's make sure that our sonoffminiR2 uses the DNS server of our local router (in my case 10.0.0.1) and not the external ones.

then from any computer on the LAN network via console we issue the commands:

- to see the sonoff info

curl -XPOST --header "Content-Type: application/json" --data-raw '{"data": {}}' http://10.0.0.118:8081/zeroconf/info

responds with information where you see the device in the otaUnlock: false state

- to unlock

curl -XPOST --header "Content-Type: application/json" --data-raw '{"data": {}}' http://10.0.0.118:8081/zeroconf/ota_unlock

the response is immediate and gives {"seq":2,"error":0}

- to flash

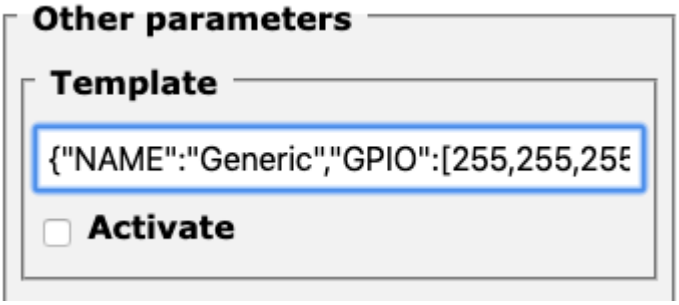
curl http://10.0.0.118:8081/zeroconf/ota_flash -XPOST --data '{"data":{"downloadUrl": "http://10.0.0.108/tasmota-lite.bin", "sha256sum": "123a378c9da7f2fdf9a4870ddecb06742cd1ab529b93ec2bb9419b88d6dc6bee"} }'

responds with {"seq":3,"error":0}
the flash is not immediate wait a minute after the response

sometimes you have to move the .htaccess file directly to the main directory /var/ww/html and then reposition the outlet in v2. This is because you would need to change something in the apache configuration but I don't feel like investigating.

Once installed Tasmota configured and updated from lite to the latest full version

go to**Configuration**" > "**Configure other**".Above, under the voice "**Other parameters**" > " **Templates**", in the form



we can insert according to the type of sonoffminir2

{"NAME":"Sonoff Mini","GPIO":[17,0,0,0.9,0,0,0,21,56,0,0,255],"FLAG":0,"BASE": 1} or

{"NAME":"Sonoff Mini","GPIO":[17,0,0,0,9,0,0,0,21,158,0,0,255],"FLAG":0,"BASE":1}

Once you have finished editing, click on "**Activate**" and then, at the bottom, on "**Save**". The unit will reboot and function as it should.

Below is the explanation from the tasmota website for this peculiarity:

Configuration (old format, will be converted to new template when applied)

{"NAME":"Sonoff Mini","GPIO":[17,0,0,0.9,0,0,0,21,56,0,0,255],"FLAG":0,"BASE": 1}

Alternative template where the blue LED lights up only in case of connection issues and on button actions:

{"NAME":"Sonoff Mini","GPIO":[17,0,0,0,9,0,0,0,21,158,0,0,255],"FLAG":0,"BASE":1}

The Sonoff Mini is, as the name implies, a very small device, designed to be hidden where there's only supposed to be cables.
Since it's not supposed to be visible, it's also designed to easily attach an external button.