Pi.Alert

Überwacht das lokale Netzwerk (W-LAN und LAN) und speichert alle erkannten Geräte und meldet diese, wenn gewünscht, per Mail. Die erkannten Geräte werden in einer Web-Oberfläche dargestellt und können dort verwaltet werden.

Installation & Konfiguration

Installation mit Docker und wichtige Konfigurationen nach der Installation.

Installation

Mit Docker ist Pi.Alert schnell und einfach ausgerollt. Dazu einfach die folgende Compose Config anpassen.

```
services:
  pialert:
    image: jokobsk/pi.alert
    container_name: pialert
    volumes:
      - /pfad/zur/pialert/config:/home/pi/pialert/config
      - /pfad/zur/pialert/db:/home/pi/pialert/db
    environment:
      - TZ=Europe/Berlin
      - HOST_USER_ID=1000
      - HOST_USER_GID=1000
      - PORT=20211
    network_mode: 'host'
    restart: 'unless-stopped'
```

Service Name und Container Name können angepasst werden. Wichtig ist, dass die Volumes angepasst werden, je nachdem in welchem Pfad die Config und DB gespeichert werden sollen.

In der Config werden mit HOST_USER_ID und HOST_USER_GID die ID und Gruppe angepasst, unter der die Dienste in dem Container ausgeführt werden. Deswegen müssen die Berechtigungen/Eigentümer der beiden Ordner, in denen die Config und DB liegen, angepasst werden. Hierzu die beiden folgenden (an die eigenen Pfade angepassten) Befehle ausführen:

```
sudo mkdir /pfad/zur/pialert/config
sudo chown -r 1000 /pfad/zur/pialert/config
sudo mkdir /pfad/zur/pialert/db
sudo chown -r 1000 /pfad/zur/pialert/db
```

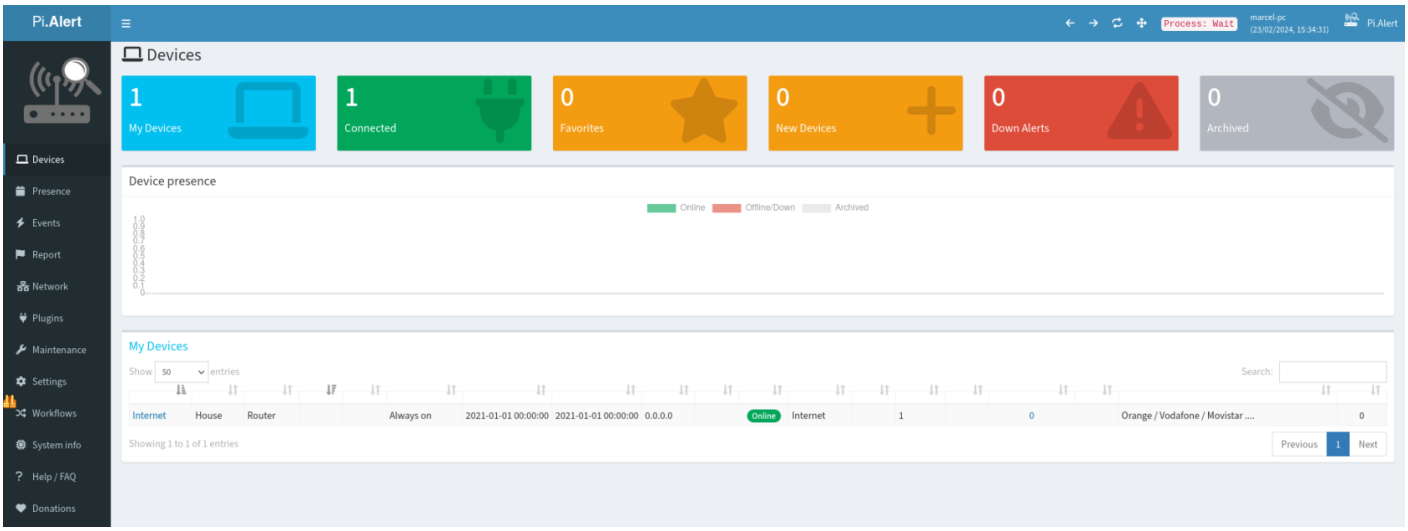
Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegeben Optionen. Sollten die Container bereits mit dieser

Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

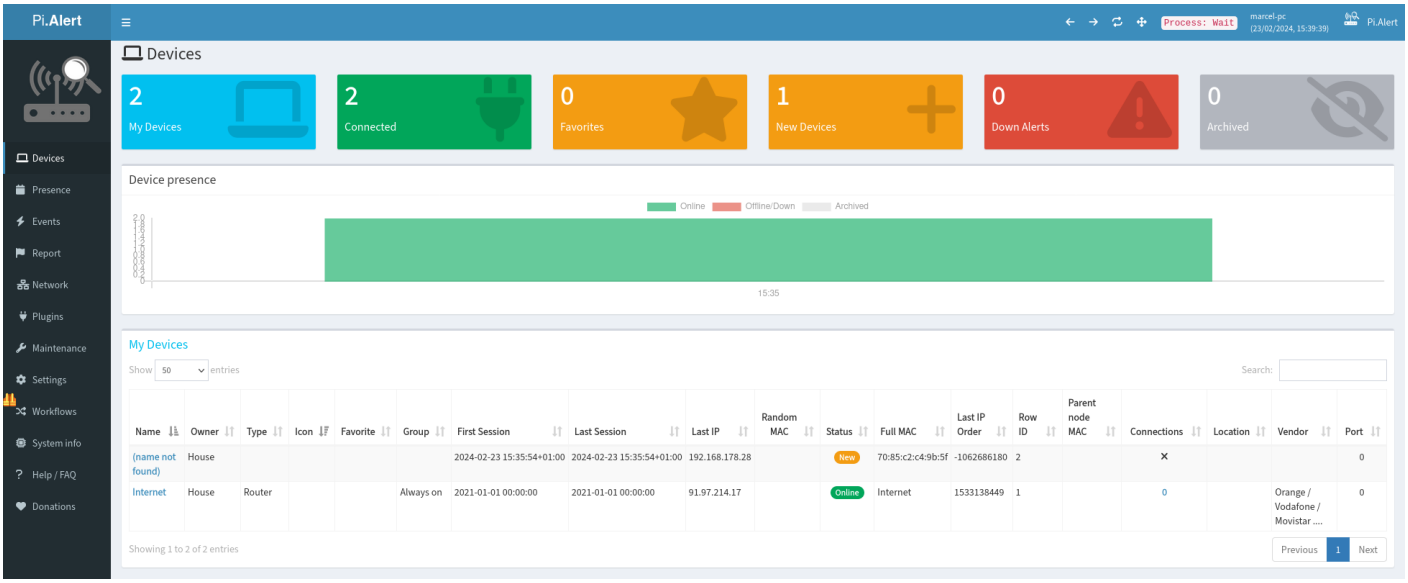
Nach wenigen Sekunden ist die Weboberfläche unter *http://Server-IP:20211* erreichbar.

Arp-Scan

Nach der Installation sind noch Konfigurationen notwendig, damit gescannt wird. Standardmäßig sind zwar ein paar Scans aktiv, aber gerade der ARP-Scan ist beim Docker Container falsch eingestellt. Außerdem ist die Seite nicht mit einem Login geschützt, sodass jeder im Netzwerk auf diese zugreifen kann. Nach dem 1. Start sieht die Seite wie folgt aus.



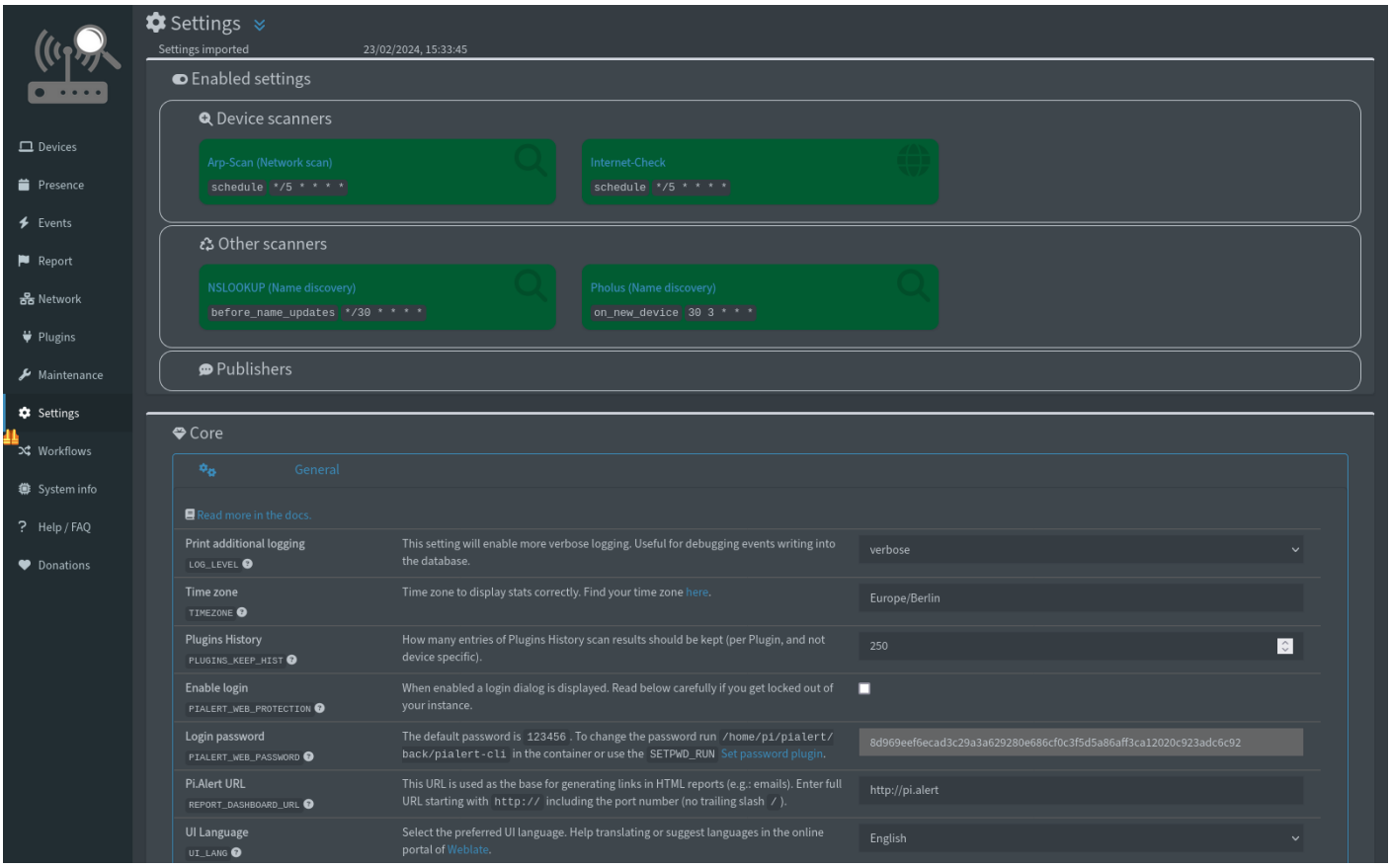
Am besten noch ein paar Minuten abwarten, da standardmäßig die ersten Scans laufen. Nach wenigen Minuten sollten dann 2 Geräte in der Liste auftauchen, das Gerät, auf dem Pi.Alert läuft und ein Gerät mit dem Namen *Internet*, der die aktuelle externe IP darstellt. die Ansicht sieht dann in etwa wie folgt aus.



Arp-Scan

Wie zu sehen ist, werden die Geräte aus dem lokalen Netzwerk nicht angezeigt. Somit wird zuerst der ARP-Scan konfiguriert. Dazu in die *Settings* wechseln. Ganz oben auf der Seite der Einstellungen werden die aktivierten Scans angezeigt.

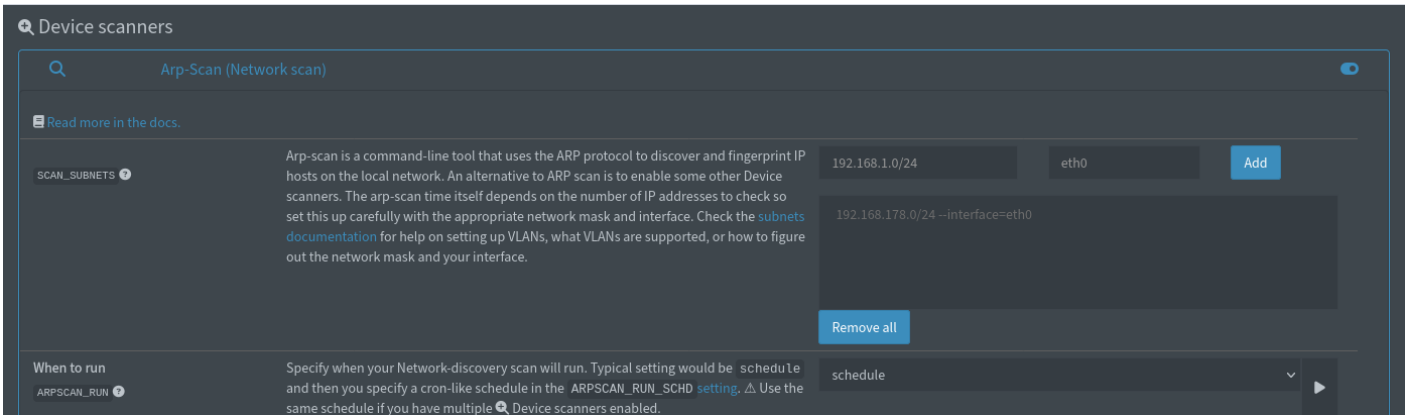
> Leider sind diese aufgrund des Standard Designs schlecht lesbar, dieses lässt sich in den im Menü *Maintenance* anpassen. Daher wird für die nachfolgenden Screenshots das Design in den Dark Mode gewechselt: Maintenance -> Toggle Modes (Dark/Light)



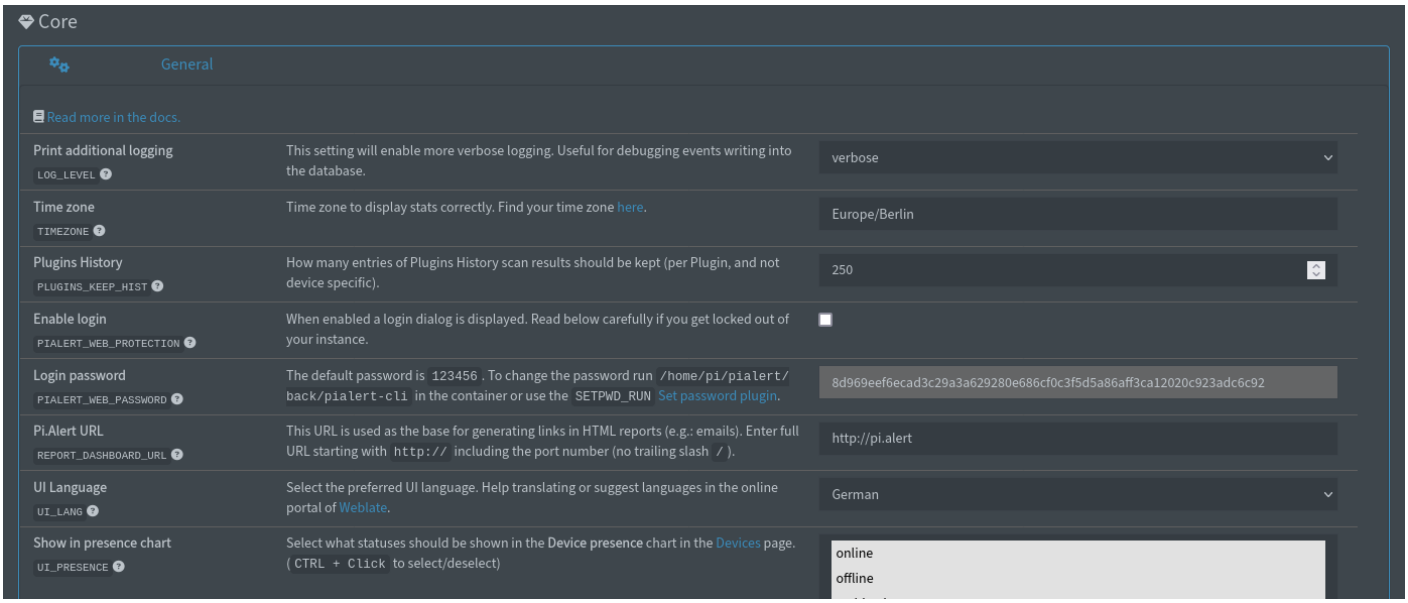
Dort ist bereits der Arp-Scan gelistet. Wenn dieser angeklickt wird, springt der Browser direkt an die passende Stelle in der Konfiguration. Dort ist dann das Problem zu sehen, es wird standardmäßig nur das Netzwerk *192.168.1.0/24* gescannt. Die meisten Heimnetzwerke verwenden häufig 192.168.178.0/24 (Standard FritzBox). Am besten einmal das eigene Netzwerk prüfen z. B. indem die Netzwerk Konfiguration auf dem eigenen Rechner geprüft wird, sofern nicht bereits bekannt. Dann mit *Remove all* das bisherige Netzwerk entfernen. Danach in das Feld darüber die eigene Netzwerk-ID eingeben (in der Regel 192.168.178.0/24 oder 192.168.0.0/24) und den Namen des Netzwerk Adapters eintragen, über den gescannt werden soll. Dieser ist in der Regel *eth0*.

> Der Name des Netzwerk-Adapters lässt sich über die *System info* einsehen. Dort dann nach *Network Hardware* suchen. Neben dem Adapter wird auch die IP von diesem angezeigt,

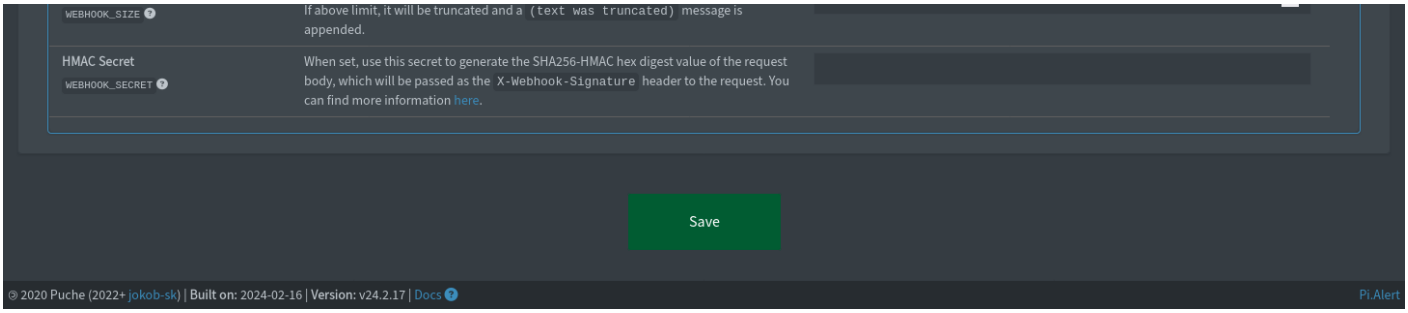
> sodass auch direkt die IP ermittelt werden kann, aus der sich dann die Netzwerk-ID ergibt (in der Regel können einfach die Zahlen nach dem letzten Punkt durch eine 0 ersetzt werden, somit wird aus 192.168.178.180/24 -> 192.168.178.0/24, die /24 bleibt selbstverständlich erhalten.



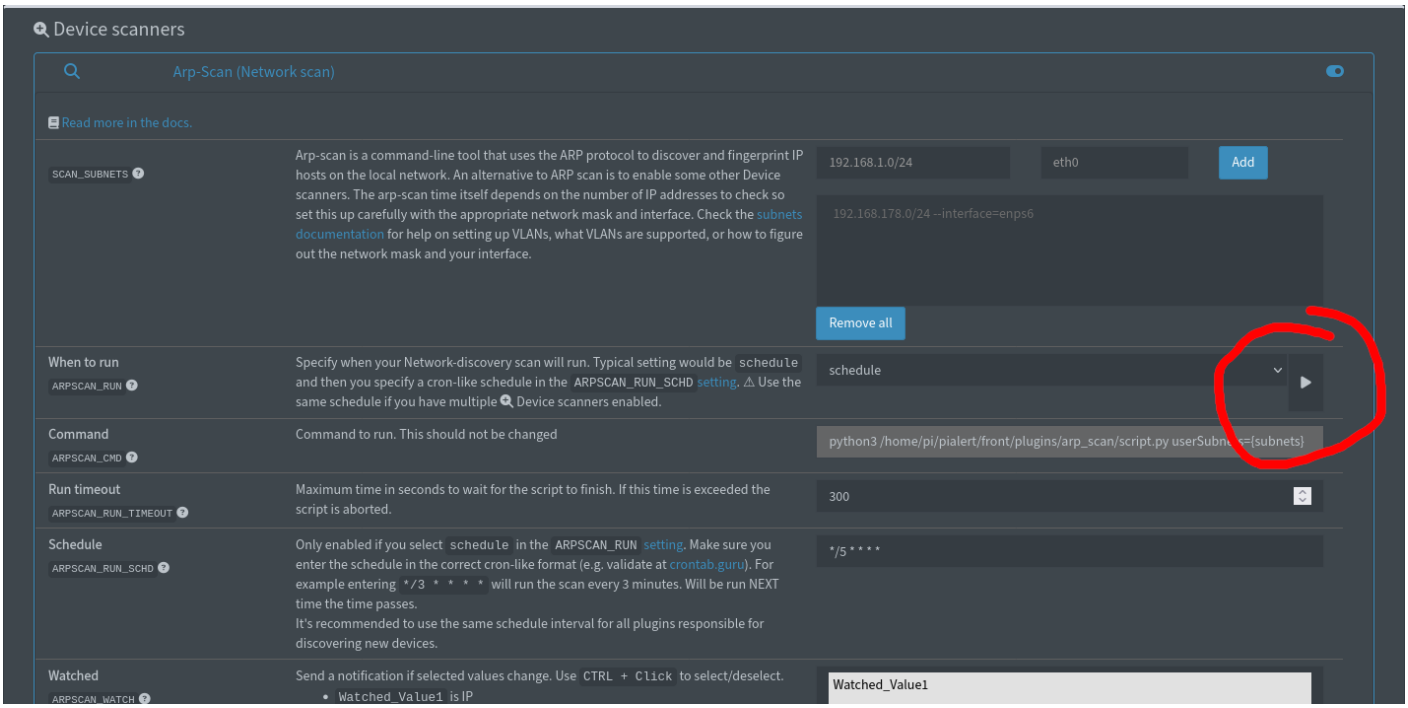
Nachdem die Einstellungen geändert wurden, könnte noch die Sprache angepasst werden, dazu nochmal nach ganz oben scrollen und unter *UI language* die Sprache ändern.



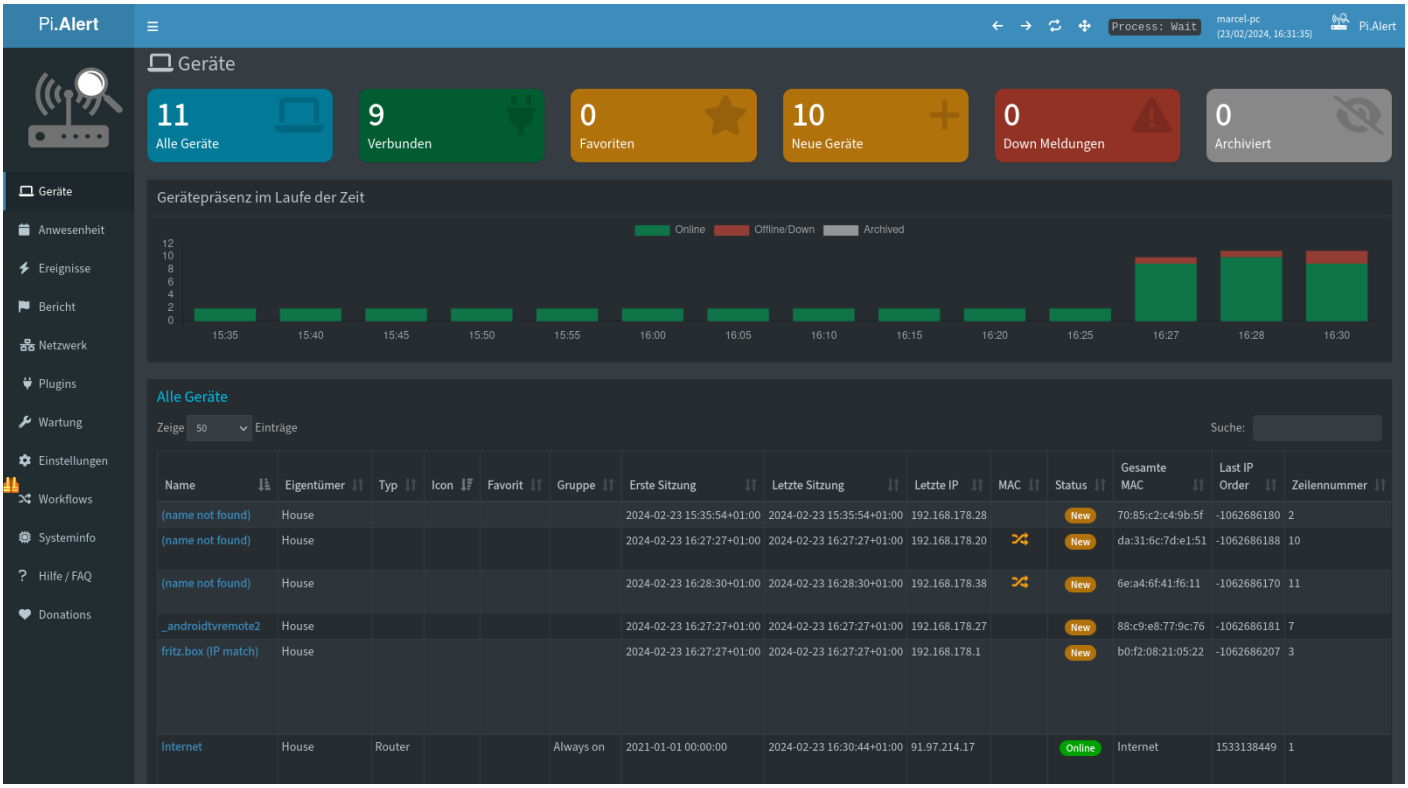
Um die vorgenommen Einstellungen zu speichern nun ans Ende der Seite scrollen und auf *Save* klicken.



Zum Schluss lädt Pi.Alert einmal neu und einige Minuten später sollten die ersten Scan Ergebnisse vorliegen. Wer nicht so lange warten möchte (alle 5 min wird standardmäßig gescannt), kann den Scan in den Einstellungen manuell anstoßen, dazu nochmal in Einstellungen (Settings) wechseln. Wieder zum Menü für den Arp-Scan wechseln und dort auf den kleinen Play-Button klicken.

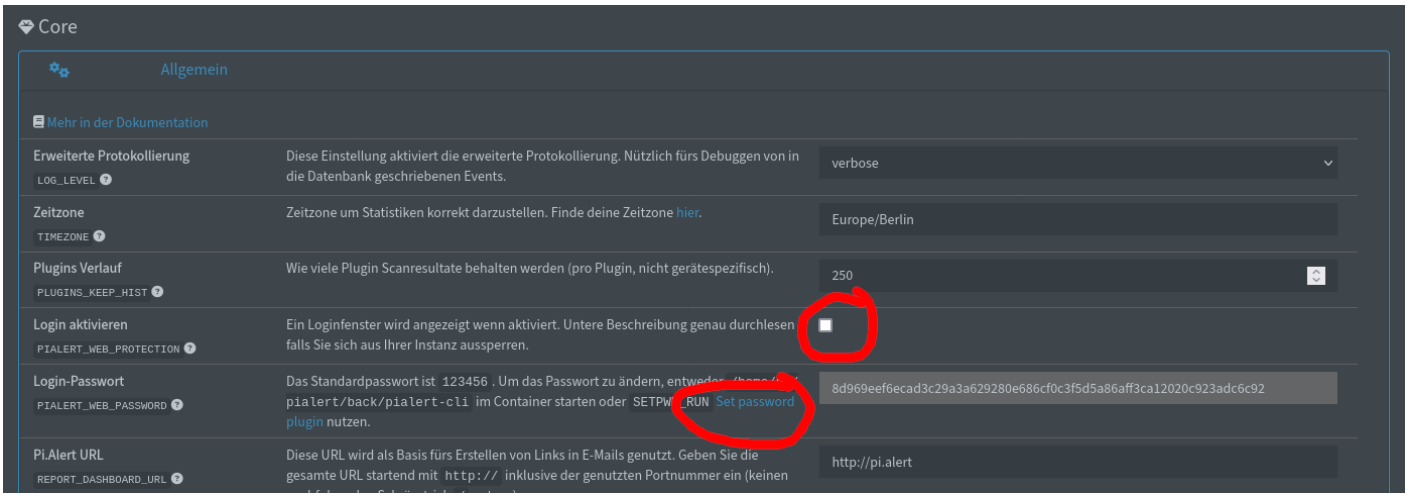


Wenn der Scan abgeschlossen ist bzw. sobald die ersten Geräte gefunden wurden, könnte es so aussehen. Nun können die Geräte bearbeitet werden, um sie mit Details wie Name, Hersteller, Icon usw. zu versehen.

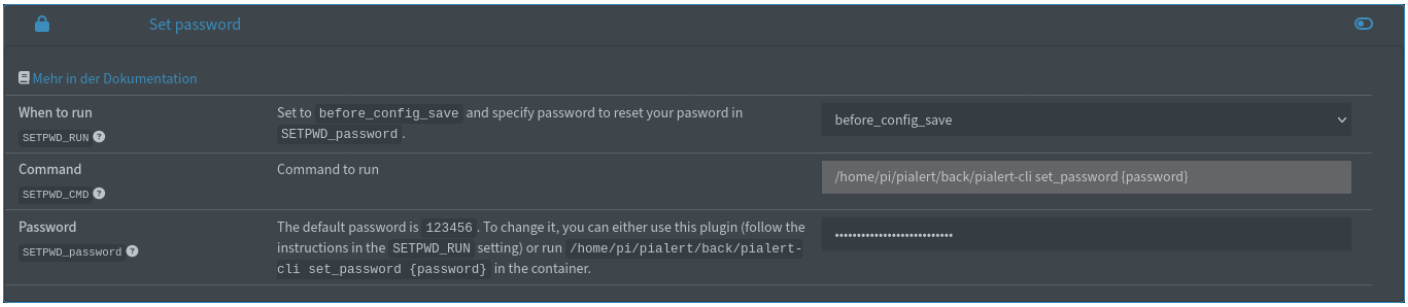


Login aktivieren

Während der Scan läuft ist die perfekte Gelegenheit, um die Seite mit einem Login zu schützen. Dazu wieder in die Einstellungen wechseln. Dort ist dann auch unter *Core* direkt die Checkbox zum Aktivieren des Logins zu sehen. Nachdem setzen des Hakens, auf den Link darunter `Set password plugin` anklicken.



Daraufhin springt der Browser zu dem Bereich *Set password*, indem nun oben erstmal *When to run* auf *before_config_save* gestellt wird und in das Feld neben *Password* wird das gewünschte Passwort eingetragen.

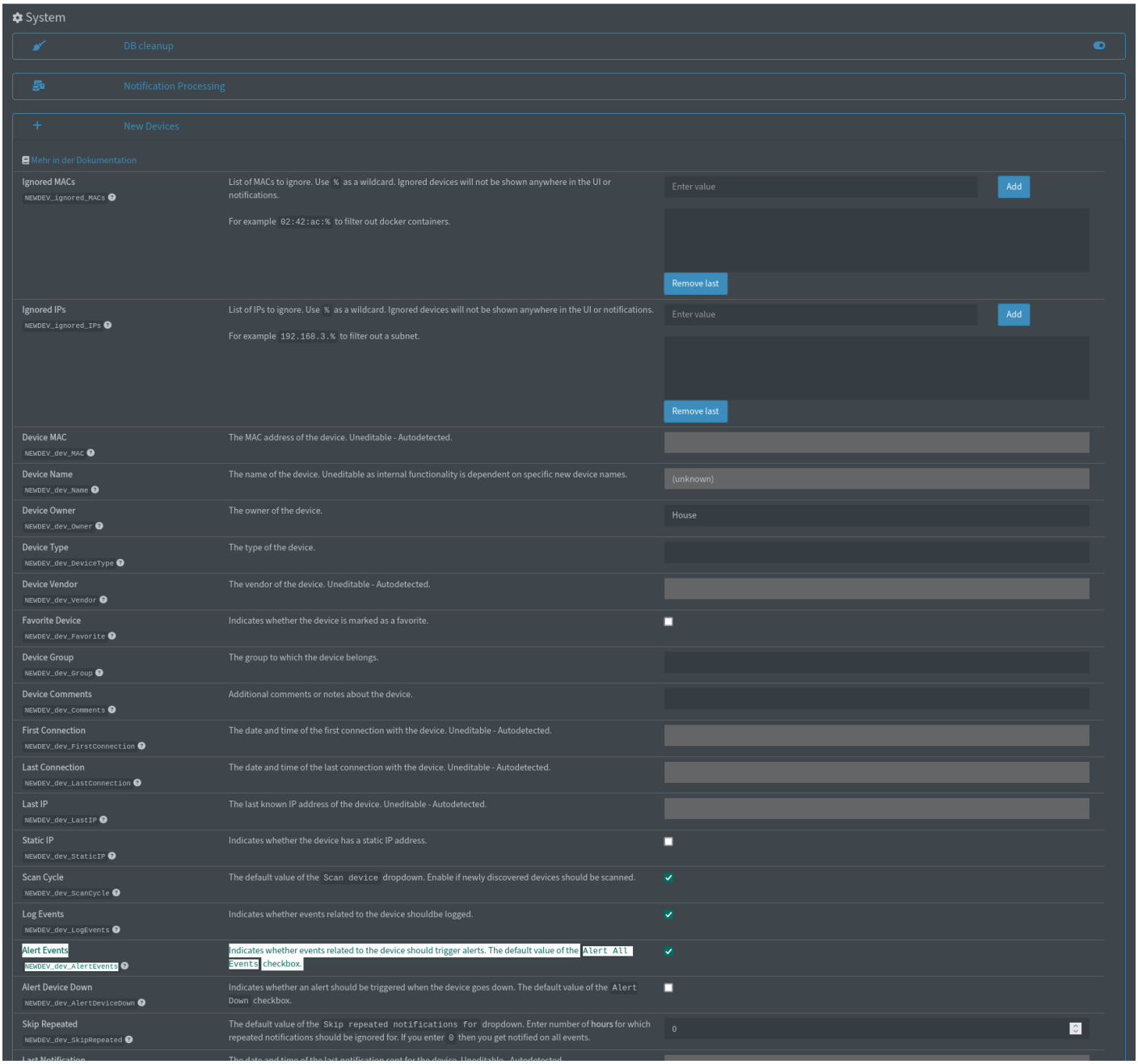


Zum Schluss wieder ganz unten auf *Save* klicken.

Automatische Up/Down Meldung deaktivieren

Standardmäßig ist eingestellt, dass für alle neuen Geräte Up/Down Meldungen berichtet werden. Das kann stören, wenn es sich um Geräte handelt, die sich regelmäßig trennen und wieder verbinden, besonders wenn Mail-Benachrichtigungen konfiguriert sind. Es wird jedes Mal eine Mail versendet, wenn Geräte sich trennen und wieder verbinden.

Um dies zu ändern, zuerst in die Einstellungen wechseln. Danach zur Rubrik *System* scrollen und dort nach der Unterrubrik *New Devices* suchen. In dieser werden alle Standard-Einstellungen hinterlegt, welche bei neuen Geräten (nicht bei bereits vorhandenen Geräten) gesetzt werden sollen. Dort gibt es eine Einstellungen mit dem Namen *Alert Events*. Wenn bei dieser der Haken entfernt wird, wird für alle neuen Geräte die Up/Down Meldung deaktiviert. Trotzdem kann sie bei Bedarf in den Einstellungen der Geräte je Gerät geändert werden, sodass für gewünschte Geräte eine Meldung verschickt wird. Unabhängig von der Meldung werden die Ereignisse weiterhin aktualisiert, um den Online Status von Geräten im Log nachschauen zu können. Im folgenden Foto ist der anzupassen Menüeintrag hervorgehoben.


System
DB cleanup
Notification Processing
New Devices
Mehr in der Dokumentation
Ignored MACs
NEWDEV_ignored_MACs
List of MACs to ignore. Use % as a wildcard. Ignored devices will not be shown anywhere in the UI or notifications.
For example 02:42:ac:% to filter out docker containers.
Enter value
Add
Remove last
Ignored IPs
NEWDEV_ignored_IPs
List of IPs to ignore. Use % as a wildcard. Ignored devices will not be shown anywhere in the UI or notifications.
For example 192.168.3.% to filter out a subnet.
Enter value
Add
Remove last
Device MAC
NEWDEV_dev_MAC
The MAC address of the device. Uneditable - Autodetected.
Device Name
NEWDEV_dev_Name
The name of the device. Uneditable as internal functionality is dependent on specific new device names.
(unknown)
Device Owner
NEWDEV_dev_Owner
The owner of the device.
House
Device Type
NEWDEV_dev_DeviceType
The type of the device.
Device Vendor
NEWDEV_dev_Vendor
The vendor of the device. Uneditable - Autodetected.
Favorite Device
NEWDEV_dev_Favorite
Indicates whether the device is marked as a favorite.
Device Group
NEWDEV_dev_Group
The group to which the device belongs.
Device Comments
NEWDEV_dev_Comments
Additional comments or notes about the device.
First Connection
NEWDEV_dev_FirstConnection
The date and time of the first connection with the device. Uneditable - Autodetected.
Last Connection
NEWDEV_dev_LastConnection
The date and time of the last connection with the device. Uneditable - Autodetected.
Last IP
NEWDEV_dev_LastIP
The last known IP address of the device. Uneditable - Autodetected.
Static IP
NEWDEV_dev_StaticIP
Indicates whether the device has a static IP address.
Scan Cycle
NEWDEV_dev_ScanCycle
The default value of the Scan device dropdown. Enable if newly discovered devices should be scanned.
Log Events
NEWDEV_dev_LogEvents
Indicates whether events related to the device shouldbe logged.
Alert Events
NEWDEV_dev_AlertEvents
Indicates whether events related to the device should trigger alerts. The default value of the Alert All Events checkbox.
Alert Device Down
NEWDEV_dev_AlertDeviceDown
Indicates whether an alert should be triggered when the device goes down. The default value of the Alert Down checkbox.
Skip Repeated
NEWDEV_dev_SkipRepeated
The default value of the Skip repeated notifications for dropdown. Enter number of hours for which repeated notifications should be ignored for. If you enter 0 then you get notified on all events.
0

Funktionsübersicht

Auf dieser Seite soll das Tool kurz vorgestellt werden. Was macht es und wie sieht es aus.

Der offizielle GitHub von Pi.Alert: https://github.com/pucherot/Pi.Alert

Pi.Alert scannt permanent das lokale Netzwerk und erkennt neue Geräte und versucht möglichst viele Informationen über diese Geräte zu ermitteln. Unter anderem wird geprüft, ob Geräte dauerhaft online sind, dies wäre z. B. bei einem Drucker der Fall. Außerdem erkennt das Gerät, wenn ein Gerät eine neue IP-Adresse erhält.

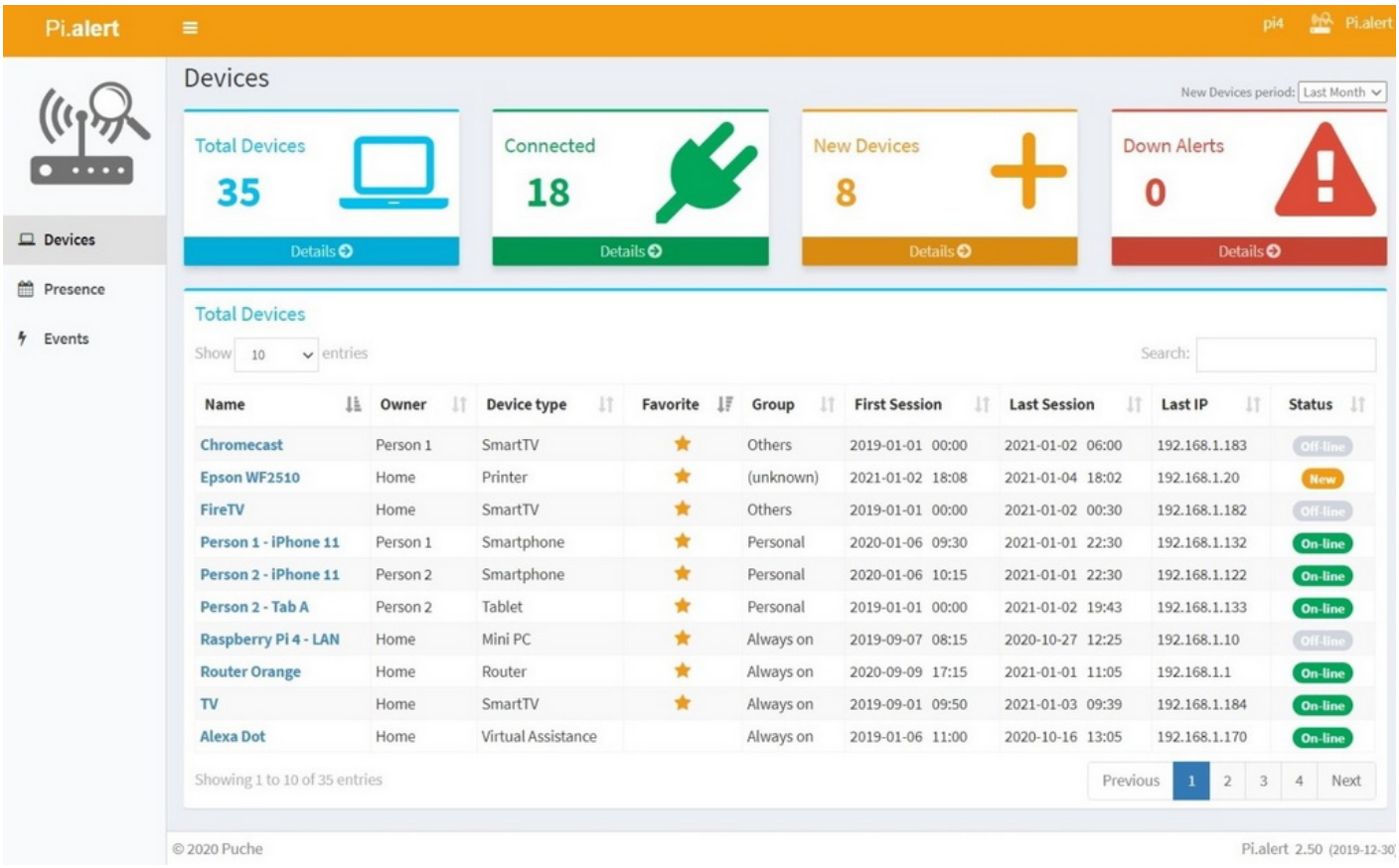
Die erkannten Geräte und ihr Status sowie die Informationen über diese werden in einer Webübersicht dargestellt. Selbstverständlich können die Geräte ähnlich einem Inventar verwaltet und mit zusätzlichen Informationen versehen werden. So können bekannte Geräte mit einem Namen versehen werden, um neue unbekannte Geräte noch leichter und schneller zu erkennen.

Folgende Scan Methoden unterstützt Pi.Alert:

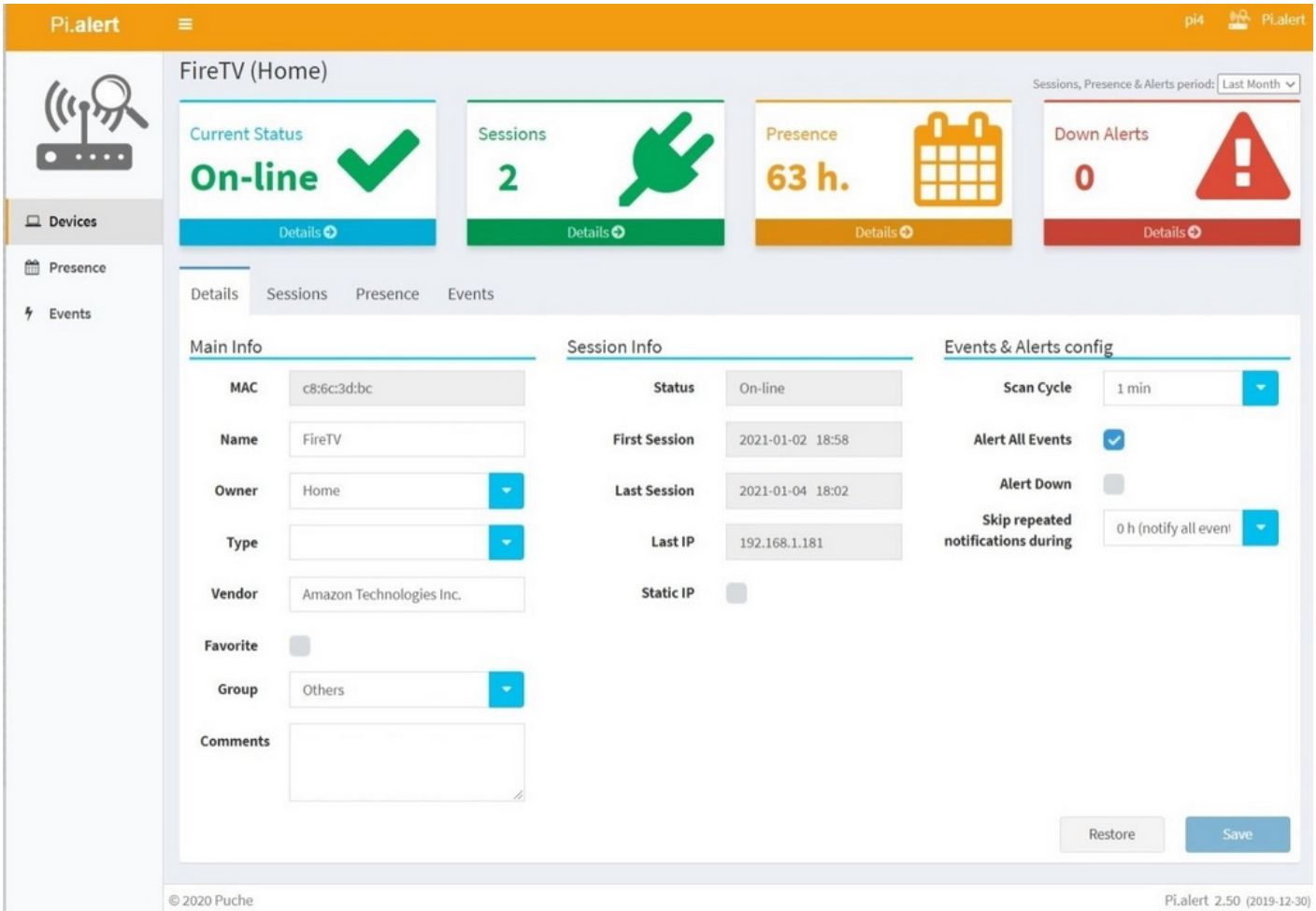
- ARP: Damit IP-Pakete im lokalen Netzwerk zugestellt werden können, müssen die Geräte vorher neben der IP auch die MAC Adresse austauschen. Dazu dient eine sog. ARP Abfrage. MIt dieser kann auch der Pi.Alert die IPs im Netzwerk den MAC Adressen zuordnen.
- Pi-Hole: Pi.Alert kann einen Pi-hole anzapfen, da dieser sowieso Geräte kennt, die diesen als DNS-Server nutzen. Aus den registrieren DNS-Clients kann der Pi.Alert dann die Geräte im Netzwerk ermitteln (jedoch könnten dadurch Geräte "durchs Netz rutschen").
- Dnsmasq: Genauso wie beim Pi-hole kann auch der DNS- und DHCP-Server Dnsmasp zum identifizieren der Geräte genutzt werden.

Hier ein paar Screenshots aus der offiziellen Dokumentation:

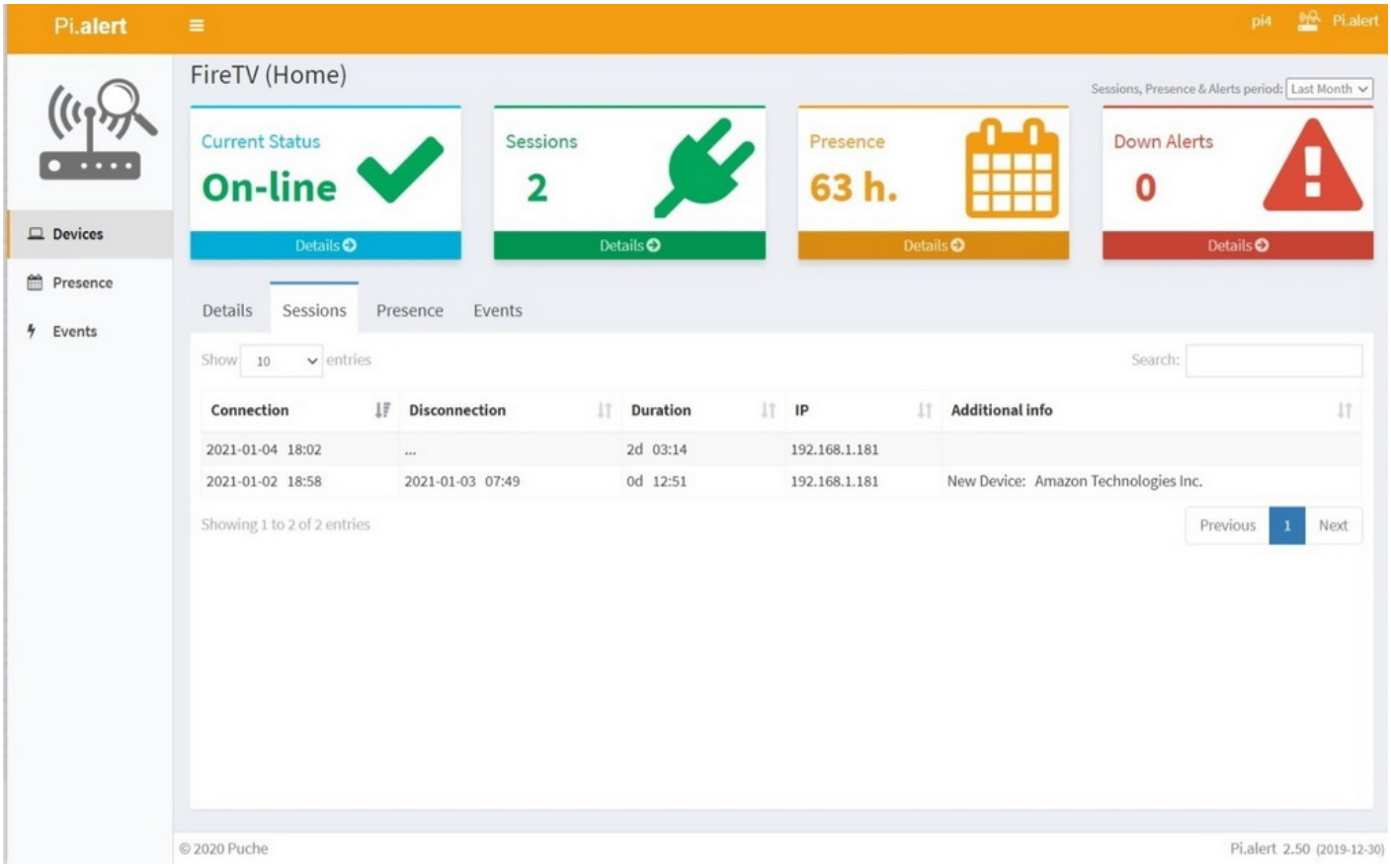
Dashboard:



Details eines Gerätes:



Sitzungen eines Gerätes:



Anwesenheit eines Gerätes: