

Scanner et trier des documents avec une télécommande Philips Hue

Détournement d'une télécommande Philips Hue pour lancer un scan automatique qui finit sa course dans Paperless-ngx

- [Scanner d'une simple pression](#)
- [Préparer le nid](#)
- [Le script de numérisation](#)

Scanner d'une simple pression

On connaît tous cette "pile de papier" qui traîne dans une banette ou dans un coin du bureau. Celle où l'on se dit qu'on va la traiter "plus tard" car il faut lancer le pc, allumer l'imprimante, lancer un logiciel pour scanner, enregistrer le PDF, le renommer pour qu'il soit reconnaissable, et blablabla.



J'ai décidé de régler ce problème avec une solution radicale : transformer une télécommande Philips Hue en baguette magique.

J'ai fait en sorte que tout simplement, avec une longue pression sur l'un des boutons, le scan se lance automatiquement pour apparaître, trié et indexé dans ma bibliothèque numérique.

Voici comment j'ai fait



Les acteurs de la pièce

Pour ce projet, j'ai utilisé toute la force de mon homelab. Chaque machine a un rôle précis, comme une équipe bien huilée

- **Le cerveau (HP EliteDesk Mini)** : Il fait tourner Home Assistant, c'est lui qui va écouter les signaux de la télécommande Philips Hue et qui va donner l'ordre de mission
- **Les Bras (Un autre EliteDesk Mini)** : Lui, il me sert habituellement comme serveur Multimédia, il tourne sous Debian 12 et héberge une instance CasaOS. Mais du coup, c'est lui qui va recevoir le signal de Home Assistant, lancer le scan, et générer le fichier
- **La Sagesse (Mac Mini M1)** : Il héberge Paperless-ngx. C'est lui qui va recevoir le fichier, lire le texte (OCR) et ranger le document au bon endroit grâce à ses algorithmes.

Préparer le nid

Avant que le CasaOS ne puisse rapporter ses trouvailles, il lui faut un endroit où les déposer. Dans **Paperless-ngx**, cet endroit s'appelle le dossier "**Consume**". C'est un dossier que Paperless surveille 24h/24 : dès qu'un fichier tombe dedans, il l'attrape, l'analyse et le range.

Créer un point de chute

Dans mon installation, le dossier "**Consume**" n'est pas sur le Mac Mini, ni sur le serveur de scan. Il se situe dans un NAS.

J'ai fait ce choix pour que le dossier consume soit accessible facilement dans un point de stockage déjà connu de mes machines.

Sur mon NAS, j'ai créé un dossier "Aspirateur_de_documents" qui sera mon dossier consume.



J'ai ensuite monté ce dossier du NAS dans un répertoire local (`/mnt/aspirateur_de_documents`) sur mon **CasaOS** grâce au fichier `/etc/fstab`, comme ça en plus, il se monte automatiquement au démarrage.

J'ai fait également la même chose sur le **Mac Mini** qui héberge **Paperless-ngx** pour qu'il surveille ce dossier.

Le réglage crucial ici, c'est le protocole de partage, le dossier est partagé en SMB, le serveur **CasaOS** a les droits en écriture et le **Mac Mini** lui a les droits en écriture, mais aussi en suppression, sinon il ne pourrait pas supprimer le fichier qu'il a importé et essaierait de l'importer en boucle (le pauvre).

Configuration de Paperless-ngx

C'est ici, sur mon **Mac Mini**, que réside l'intelligence du système, Paperless-ngx est prêt, mais il faut lui expliquer où se trouve sa "gamelle" (le fameux dossier **Aspirateur_de_documents** sur le NAS) et comment il doit se comporter.

Le branchement vers Docker

Comme la plupart des installations autohébergées, mon Paperless tourne dans un container Docker

Pour qu'il puisse "voir" le dossier du NAS, j'ai dû créer un pont dans mon fichier de configuration (`docker-compose.yml`).

Dans la section `volumes`, j'ai ajouté cette ligne :

- `/Volumes/aspirateur_de_documents:/usr/src/paperless/consume`

C'est comme si je disais à Paperless : "Ce dossier que tu vois sur le réseau, considère-le comme ton dossier d'entrée".

Les ordres de mission (Variables d'environnement)

Pour que le traitement soit aux petits oignons, j'ai ajusté quelques réglages dans le fichier de configuration de Paperless:

`PAPERLESS_CONSUMER_POLLING=60` : C'est mon réglage préféré. Comme le dossier est sur un NAS (en réseau), les notifications automatiques de fichiers ne marchent pas toujours bien. En mettant "60", Paperless va toquer à la porte du NAS toutes les minutes pour voir s'il y a du nouveau. C'est le secret de la réactivité !

`PAPERLESS_OCR_LANGUAGE=fra` : On n'oublie pas de lui dire de lire le français, sinon il va bégayer sur nos accents.

La magie du tri automatique

C'est là que Paperless devient incroyable. Une fois qu'il a "aspiré" le PDF envoyé par le serveur CasaOS (ou mis dans le NAS manuellement), il ne se contente pas de le stocker.



1. **Il va lancer un OCR (Reconnaissance Optique de Caractères)** : Il transforme l'image du scan en texte recherchant.
2. **Il analyse** : S'il voit des mots comme "Facture" et "EDF", il va appliquer tout seul les tags que j'ai pré-configuré (Par exemple le tag "Facture", "Appartement", "Électricité"...)
3. **Il archive** : Il va renommer mon fichier proprement et le ranger dans sa base de données

À ce stade, toute la logistique de réception est en place. Le **NAS** sert de boîte aux lettres intelligente et **Paperless-ngx** guette patiemment, prêt à bondir sur le moindre PDF qui se trouvera dans le dossier.

Mais pour l'instant, notre "Aspirateur de documents" est bien vide...

Il est temps de passer à l'action et de voir comment on va faire pour envoyer des documents dans ce dossier.

Le script de numérisation

Maintenant que notre "Aspirateur de documents" est prêt sur le NAS et sur Paperless-ngx, il faut que notre **CasaOS** sache comment attraper la feuille de papier et l'envoyer dans les tuyaux. C'est ici que l'on entre en salle des machines.

Dompter le matériel avec SANE



Pour piloter un scanner en ligne de commande Linux, on utilise l'outil **SANE** (Scanner Access Now Easy). C'est lui qui va faire le pont entre le logiciel et le scanner

D'abord, on va installer les utilitaires de base :

```
sudo apt update && sudo apt install sane-utils
```

Ensuite, on vérifie que le serveur "voit" bien le scanner (ici, il est en réseau) avec la commande

```
scanimage -L
```

```
renard@casaos:~$ scanimage -L  
device `escl:https://10.x.x.x:443' is a Canon TS3300 series platen scanner
```

Le script magique

J'ai écrit un petit script Bash très simple. Son rôle ? Réveiller le scanner, scanner le document posé sur la vitre et l'envoyer fissa dans notre dossier monté sur le NAS.

Voici à quoi il ressemble :

```
#!/bin/bash  
# On définit le nom du fichier avec la date  
DATE=$(date +%Y%m%d_%H%M%S)  
FILENAME="scan_$(date +%Y%m%d_%H%M%S).pdf"  
  
# On lance le scan  
scanimage -d "airscan:e0:Canon TS3300 series" --format=pdf --resolution 300 > /tmp/$FILENAME
```

```
# On envoie le fichier vers le dossier consume du NAS
cp /tmp/$FILENAME /mnt/aspirateur_de_documents

# On nettoie le fichier temporaire
rm /tmp/$FILENAME
```

Ensuite, j'ai rendu le script exécutable avec un petit `chmod +x scan_vers_paperless.sh`

Le chef d'orchestre (Home Assistant & SSH)

C'est ici, sur mon **Home Assistant** que la magie se déclenche. Home assistant doit pouvoir dire, quand j'appuie sur la télécommande au serveur CasaOS : "Hey, lance le scan maintenant !".



Le "badge d'accès" (La Clé SSH)

Pour que Home Assistant puisse commander CasaOS sans me demander mon mot de passe à chaque fois, j'ai créé un pont sécurisé avec une **clé SSH**.

Depuis le terminal de Home Assistant, on génère une clé et on l'envoie sur CasaOS

```
ssh-copy-id -i /config/.ssh/id_rsa.pub renard@casaos
```

Désormais, Home Assistant peut se connecter sur le CasaOS sans souci

Déclarer la commande dans HA

Dans mon fichier `configuration.yaml`, j'ai ajouté cette ligne pour transformer mon script en une action utilisable par HA :

```
shell_command:
  declencher_scan_paperless: "ssh -i /config/.ssh/id_rsa -o StrictHostKeyChecking=no
renard@casaos 'bash /home/renard/scan_vers_paperless.sh'"
```

L'automatisation : La télécommande magique

Enfin, la touche finale. J'ai créé une automation toute simple avec ma télécommande **Philips Hue** déjà reliée a mon Home Assistant :

- **Déclencheur** : Un appui long sur la touche du bas de la télécommande
- **Action** : Appeler le service `shell_command.declencher_scan_paperless`.

Conclusion

Maintenant quand je reçois un document que je dois scanner, j'ai juste à poser la feuille sur la vitre du scanner, j'appuie longuement sur la télécommande qui est à côté contre le mur, et je continue ma journée.

Je sais que quelques minutes plus tard, je retrouverai mon document parfaitement nommé et tagué dans **Paperless-ngx**.

C'est ça, la magie de l'auto-hébergement : faire travailler nos machines ensemble pour nous libérer l'esprit.