

260611_procédure Komga-PDF par Claude sur StirlingPDF

Présentation

La procédure Komga-PDF est un script de compression de fichiers PDF pour réduire considérablement la taille du stockage des bibliothèques d'ouvrages de la collection de Julien.

L'acquisition de fichiers issus de bases de données Internet accumule des fichiers PDF en haute résolution (images surtout). Ces fichiers sont synchronisés entre les NAS Synology et un disque dur kDrive Infomaniak de 6 To.

Contexte

Des points de montage Rclone relient les répertoires kDrive et les containers Docker installés sur le VPS OVH juxjux.ovh. Ces containers lisent les PDF à la volée — permettant ainsi de construire une infrastructure de très forte densité de connaissances digitales (epub, pdf, images, vidéos, musiques...) avec une solution serveur d'entrée de gamme (10 €/mois tout inclus).

La lecture réseau est multi-support : PC, tablette, téléphone. Elle est aussi nomade, séquentielle et pressée. La compacité des fichiers PDF devient donc nécessaire pour diffuser rapidement sur les réseaux. Un magazine ou un ouvrage informatique ne doit pas peser 130 Mo sur un écran de 10 pouces.

Procédure

- Les nouveaux PDF sont déposés dans le dossier Syncthing : `Syncthing/komga-pdf/` depuis n'importe quelle machine (PC Elio+Jux, PC Nexte, Ubuntu, Xiaomi...)

- Syncthing synchronise automatiquement vers le VPS : `/home/debian/Documents/komga-pdf/`
- Le service `komga-watch` (systemd VPS) détecte l'arrivée via inotifywait local
- Dès détection, `komga_compress_vps.py` compresse le PDF via StirlingPDF et dépose le résultat dans `/home/debian/Documents/komga-compressed/`
- Julien range manuellement les PDF compressés vers sasnexe puis dans le dossier thématique de son choix
- Les originaux ne sont jamais supprimés du VPS — Julien valide et conserve si besoin (haute résolution souhaitée)

Architecture technique

Élément	Chemin / URL
Dépôt (toutes machines)	<code>Syncthing/komga-pdf/</code>
Dossier VPS (Syncthing)	<code>/home/debian/Documents/komga-pdf/</code>
Destination VPS	<code>/home/debian/Documents/komga-compressed/</code>
API compression	<code>https://spdf.juxjux.ovh/api/v1/misc/compress-pdf</code>
Scripts VPS	<code>/home/debian/komga_compress_vps.py</code> + <code>/home/debian/komga_watch_vps.sh</code>
Logs	<code>journalctl -u komga-watch -f</code> (sur le VPS)

Scripts

Le service tourne entièrement sur le VPS — plus de dépendance Ubuntu ni de SSH persistant.

Fichier	Rôle
<code>komga_compress_vps.py</code>	Compression : liste les PDFs locaux, envoie à StirlingPDF, écrit dans <code>komga-compressed/</code> . Idempotent (ignore les fichiers déjà compressés).
<code>komga_watch_vps.sh</code>	Surveillance : traite le backlog au démarrage, puis boucle inotifywait locale.
<code>/etc/systemd/system/komga-watch.service</code>	Service system (pas user), actif au boot, <code>Restart=always</code> .

Gérer le service (VPS)

```
sudo systemctl status komga-watch
sudo systemctl restart komga-watch
journalctl -u komga-watch -f
```

Retour d'expérience (2026-06-13)

— Refactorisation VPS

Problème de l'architecture Ubuntu : le service tournait sur Ubuntu avec un SSH persistant vers le VPS. Deux défauts structurels :

- `inotifywait` ne détecte pas les fichiers déjà présents au redémarrage du service (backlog silencieux)
- Impossible à contrôler depuis Windows/Claude Code

Solution : service systemd sur le VPS lui-même. `inotifywait` local, pas de SSH persistant. Le script traite le backlog à chaque démarrage avant d'entrer dans la boucle de surveillance.

Résultats de compression (2026-06-13) :

Fichier	Original	Compressé	Gain
Beaux Arts - Juin 2026.pdf	75.8 Mo	61.1 Mo	-19%
Connaissance des Arts - Juin 2026.pdf	57.3 Mo	35.2 Mo	-39%
La Revue du Vin de France - Juin 2026.pdf	93.3 Mo	39.9 Mo	-57%
Livre lacto fermentation.pdf	96.6 Mo	19.6 Mo	-80%
Monde Gourmand N°93 - Juin 2026.pdf	41.2 Mo	17.4 Mo	-58%

Points techniques :

- Dossier VPS avec **D majuscule** : `/home/debian/Documents/` (Syncthing sensible à la casse)
- Race condition `inotifywait/Syncthing` : attente 3s après événement, script idempotent
- Python : `/usr/bin/python3` (système, `requests` 2.28.1 disponible)
- StirlingPDF auth : JWT via `POST /api/v1/auth/login` → `session.access_token`, valable 24h
- Fichiers `sync-conflict` Syncthing filtrés automatiquement par le script

Retour d'expérience (2026-06-12)

Résultats de compression :

Fichier	Original	Compressé	Gain
Monde Gourmand N°93 - Juin 2026.pdf	42 Mo	17 Mo	-58%
Connaissance des Arts - Juin 2026.pdf	57 Mo	35 Mo	-39%

Revision #8
Created 2026-06-11 06:12:37 UTC by Julien
Updated 2026-06-13 21:47:26 UTC by Julien