Install système

- Install signal desktop on ubuntu
- Bookstack
  - Installation bookstack
  - Backuping and restoring
  - Astuces et amélioration
  - sys commande pour gérer boostack
  - Structure base et dev
  - Hacks : notify-page-updates-for-tagged-books
  - Écriture d'une nouvelle fonctionnalité : s'abonner aux notification de pages modifiées dans un livre.
- MacBoock Pro
  - Redemarrer sur clé usb
- proxy public proxy interne
- Pb réseau DELL-9010
- Mattermost
- Sympa (pas sympa du tout)
- Pb avec VS code
- Python
  - PyLogo - discussion sur le projet
  - Next to do
- debeuggage lionel

Install signal desktop on ubuntu

Signal for Debian-based Distros
NOTE: These instructions only work for 64-bit Debian-based
Linux distributions such as Ubuntu, Mint etc.

1. Install our official public software signing key:
wget -O- https://updates.signal.org/desktop/apt/keys.asc | gpg --dearmor > signal-desktop-keyring.gpg
cat signal-desktop-keyring.gpg | sudo tee /usr/share/keyrings/signal-desktop-keyring.gpg > /dev/null

2. Add our repository to your list of repositories:
echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/signal-desktop-keyring.gpg] https://updates.signal.org/desktop/apt xenial main' |\
  sudo tee /etc/apt/sources.list.d/signal-xenial.list

# 3. Update your package database and install Signal:
sudo apt update && sudo apt install signal-desktop

Bookstack

Bookstack

Installation bookstack

installation avec docker : suivre le tuto du site

Commentaires

- Difficulté pour l'accéder en réseau, et sans avoir un nom de domaine qui permette la certification
- Difficulté pour gérer les mails (envoi de mail sur événement)

Installer BookStack avec Docker

zarev3 mars 20246 min de lecture

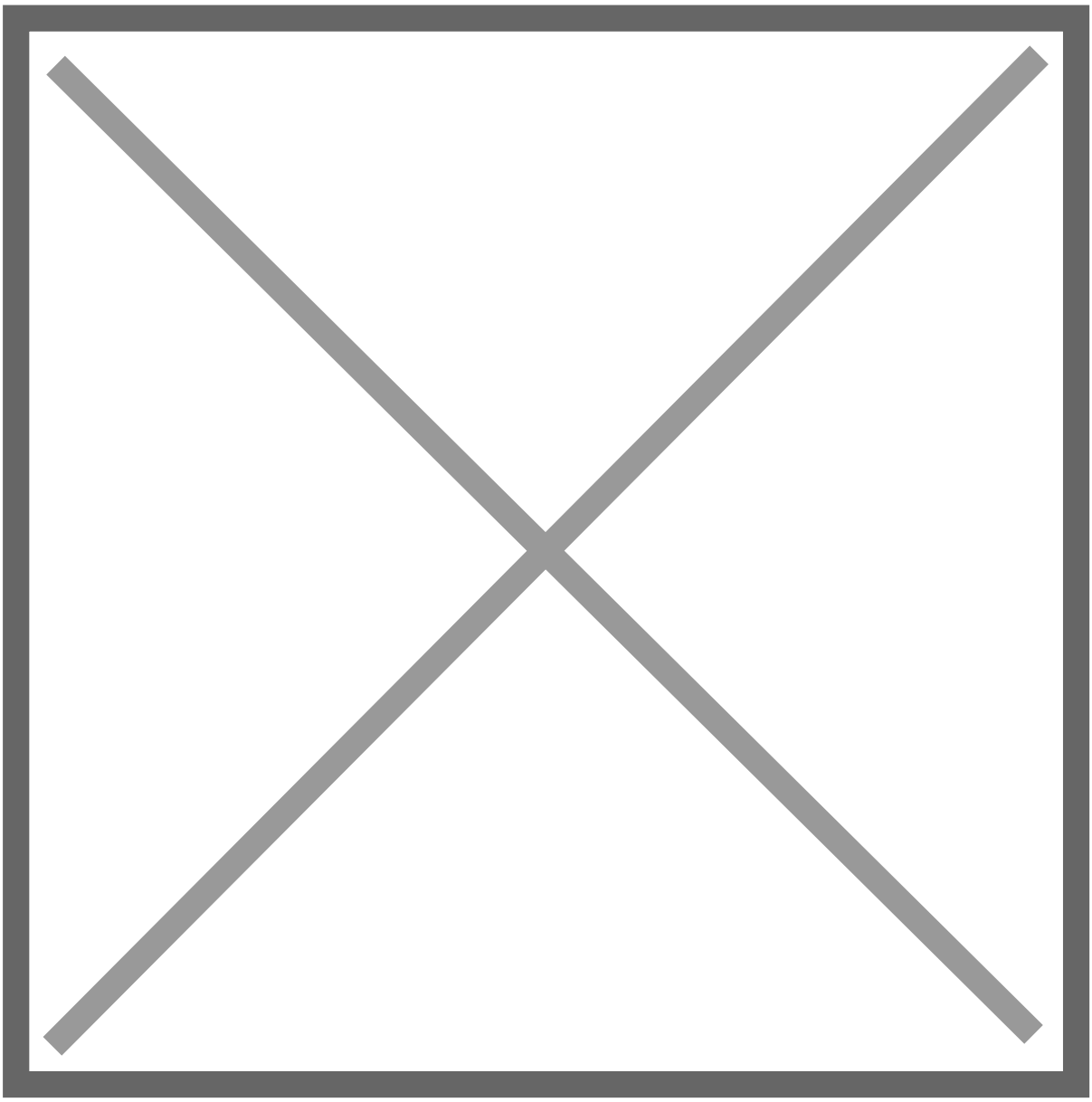
Table des matières

> Merci à **DRjekyll49** & **Biinocle** pour m'avoir averti du changement dans le compose!

BookStack est une plateforme simple, auto-hébergée et facile à utiliser pour organiser et stocker des informations sous forme de wiki.

?Caractéristiques

- Gratuit et open source: BookStack est entièrement gratuit et ouvert, sous licence MIT. La source est disponible sur GitHub. Il n'y a aucun frais pour télécharger et installer votre propre instance de bookstack.
- Interface simple et facile: la simplicité a été la priorité absolue lors de la création de BookStack. L'éditeur de page possède une interface WYSIWYG simple et tout le contenu est divisé en trois groupes simples => Livres, Chapitres, Pages.
- Consultable et connecté: le contenu de BookStack est entièrement consultable. Vous pouvez effectuer une recherche au niveau du livre ou dans tous les livres, chapitres et pages. La possibilité de créer un lien direct vers n’importe quel paragraphe vous permet de garder votre documentation connectée.
- Configurable: les options de configuration vous permettent de configurer BookStack en fonction de votre cas d'utilisation. Vous pouvez modifier le nom, le logo et les options d'enregistrement. Vous pouvez également modifier si l'ensemble du système est visible publiquement ou non.
- Exigences simples: BookStack est construit en utilisant PHP, au-dessus du framework Laravel et utilise MySQL pour stocker les données. Les performances ont été gardées à l'esprit et BookStack peut fonctionner sans problème sur un simple VPS.
- L'éditeur de pages de BookStack intègre la capacité de dessin de https://app.diagrams.net , permettant la création rapide et facile de diagrammes dans votre documentation.
- Multilingue: les utilisateurs de BookStack peuvent définir leur langue préférée. Grâce aux excellents contributeurs de la communauté, les langues actuelles intégrées à BookStack incluent EN, FR, DE, ES, IT, JA, NL, PL, RU et bien d'autres.
- Éditeur de démarques facultatif: si vous préférez écrire en Markdown alors BookStack vous soutient. Un éditeur de démarques est fourni et inclut un aperçu en direct pendant que vous rédigez votre documentation.
- Authentification intégrée: en plus de la connexion par e-mail/mot de passe par défaut, des fournisseurs sociaux tels que GitHub, Google, Slack, AzureAD et bien d'autres peuvent être utilisés. Les options Okta, SAML2 et LDAP sont disponibles pour les environnements d'entreprise.
- Fonctionnalités puissantes: en plus de la recherche et des liens puissants, il existe également un tri croisé entre les livres, des révisions de pages et une gestion des images. Un système complet de rôles et d'autorisations vous permet de verrouiller le contenu et les actions selon vos besoins.
- Authentification multifacteur: MFA est intégré et peut être appliqué au niveau de chaque rôle, si vous le souhaitez. Les options MFA incluent TOTP (Google/Microsoft Authenticator, Authy, etc...) et des codes de sauvegarde statiques.
- Modes sombre et clair: BookStack propose son interface utilisateur à la fois dans un thème clair et dans un thème sombre, sauvant ainsi les yeux de ceux qui préfèrent travailler dans l'ombre. Ceci est configurable au niveau de l'utilisateur.



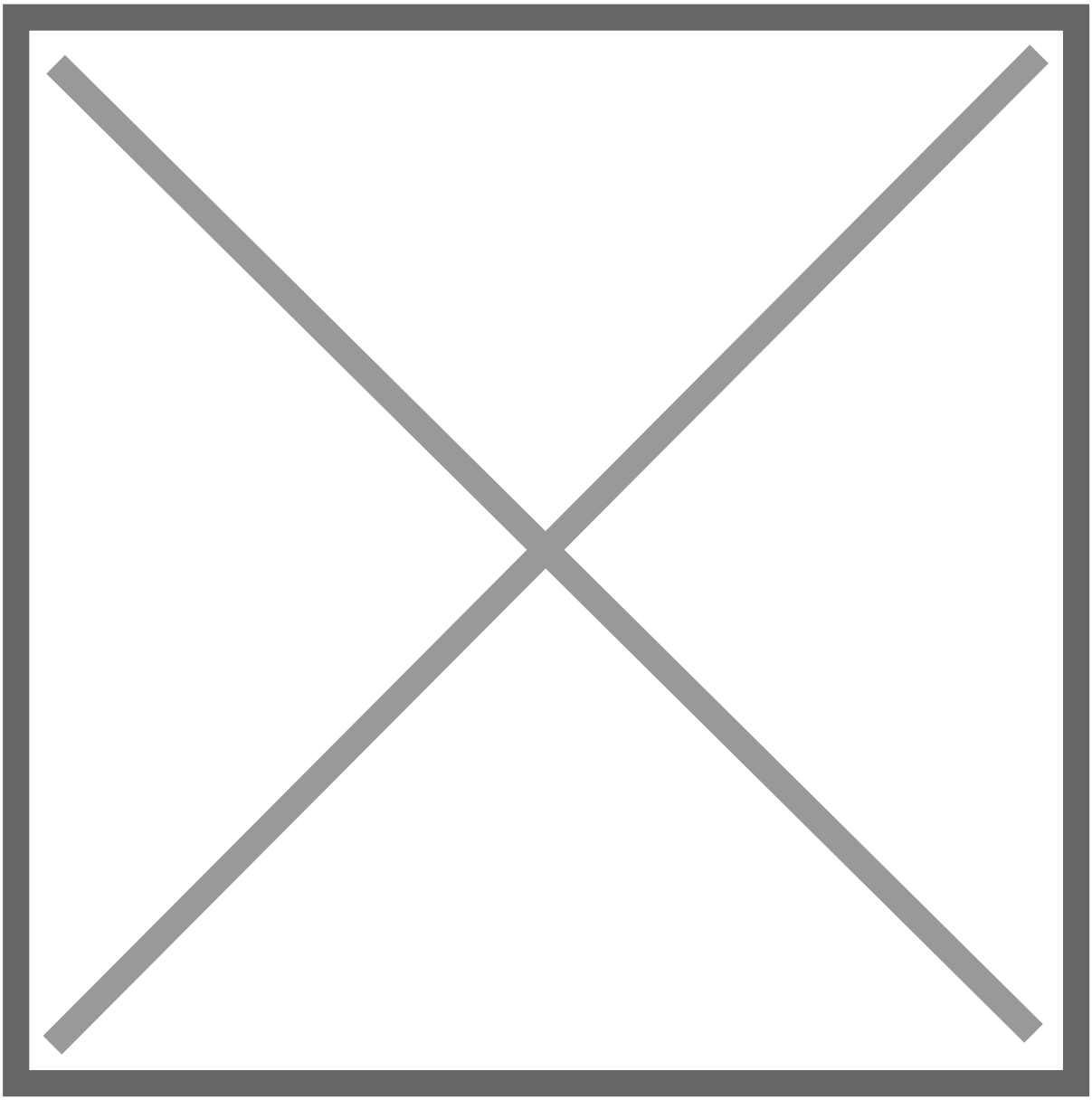
Si vous avez la moindre difficulté, n'oubliez pas que vous pouvez rejoindre belginux sur Discord et Matrix!

L'installation

1. Il faut commencer par générer une clé avec la commande suivante:

```
docker run -it --rm --entrypoint /bin/bash lscr.io/linuxserver/bookstack:latest appkey
```

Ce qui retourne:



Copiez toute la clé, base64: compris, et mettez cette clé de côté.

2. Créez votre docker-compose.yml:

```
sudo nano docker-compose.yml
```

3. Collez ça dedans en prenant soin d'adapter le port et les volumes suivant votre configuration:

```
version: "2"
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack:latest
    container_name: bookstack
```

```
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
      - APP_KEY=base64:mhoEccThD5nT/94EWcFnWQV6E8XNeNVca75J7neYz6s=
      - APP_URL=http://192.168.1.152:6875
      - DB_HOST=bookstack_db
      - DB_PORT=3306
      - DB_USERNAME=bookstack
      - DB_PASSWORD=VotreMotDePasse
      - DB_DATABASE=bookstackapp
      - APP_DEFAULT_DARK_MODE=true
      #- MAIL_DRIVER=smtp
      #- MAIL_HOST=smtp.gmail.com
      #- MAIL_PORT=465
      #- MAIL_ENCRYPTION=tls
      #- MAIL_USERNAME=yourEMAIL
      #- MAIL_PASSWORD=yourPassword
      #- MAIL_FROM=yourEMAIL
      #- MAIL_FROM_NAME=yourNAME
    volumes:
      - ./config:/config
    ports:
      - 6875:80
    restart: unless-stopped
  bookstack_db:
    image: lscr.io/linuxserver/mariadb:latest
    container_name: maria_db
    ports:
      - 3308:3306 # 3308 is a DB port visible on HOST
    environment:
      - PUID=1000
      - PGID=1000
      - MYSQL_ROOT_PASSWORD=VotreMotDePasseBis
      - MYSQL_DATABASE=bookstackapp
      - MYSQL_USER=bookstack
      - MYSQL_PASSWORD=VotreMotDePasse
    volumes:
      - ./db/config:/config
    restart: unless-stopped
```

Veillez à compléter les éléments suivants:

- ☐ **- APP_KEY=** => Indiquez la clé complète générée au début du tutoriel.
- ☐ **- APP_URL=http://192.168.1.152:6875** => soit vous indiquez votre IP locale + port ou votre nom de domaine, https.ndd.tld.
- ☐ **- DB_PASS=VotreMotDePasse** => choisissez un mot de passe fort.
- ☐ **- MYSQL_ROOT_PASSWORD=VotreMotDePasseBis** => choisissez un mot de passe fort différent de VotreMotDePasse.
- ☐ **- TZ=Europe/Brussels** => adaptez suivant votre localisation.
- ☐ **- MYSQL_PASSWORD=VotreMotDePasse** => choisissez un mot de passe fort.
- ☐ **#- APP_DEFAULT_DARK_MODE=true** => décommentez, donc retirez le signe # si vous voulez que le mode sombre soit activé par défaut.

Configurer le serveur mail

Si vous voulez activer le serveur mail (dans cet exemple, Gmail, adaptez selon le fournisseur), il faut décommenter, donc retirez le signe **#** de toutes les lignes concernant le serveur mail et compléter comme ceci:

- ☐ **MAIL_DRIVER=smtp**
- ☐ **MAIL_HOST=smtp.gmail.com**
- ☐ **MAIL_PORT=465**
- ☐ **MAIL_ENCRYPTION=tls**
- ☐ **MAIL_USERNAME=yourEMAIL** => remplacer yourEMAIL par votre Gmail.
- ☐ **MAIL_PASSWORD=yourPassword** => remplacer yourPassword par le mot de passe créé, lire l'encadré ci-dessous.
- ☐ **MAIL_FROM=yourEMAIL** => remplacer yourEMAIL par votre Gmail.
- ☐ **MAIL_FROM_NAME=yourNAME** => remplacer yourNAME par votre nom.

☐☐
Pour le mot de passe, avec Gmail c'est terminé de mettre son vrai mot de passe dans une application tierce. Ils ont mis en place un système pour créer un mot de passe différent pour une application tierce.

Je l'explique ici, à partir de **Activez la double authentification sur votre compte Gmail** jusqu'à **Copiez et conservez bien votre mot de passe, validez ensuite avec OK**.
Quand vous avez le mot de passe dédié, mettez-le dans le formulaire.

4. On va déployer l'application:

```
docker-compose up -d
```

ou avec docker-compose V2:

```
docker compose up -d
```

Cela peut prendre un peu de temps avoir que tout soit bien déployé, soyez patient.

5. Rendez-vous sur l'ip:port, suivant l'IP de votre serveur local et du port choisi, 6875 par défaut:

```
http://ip:6875/
```

Ou votre domaine:

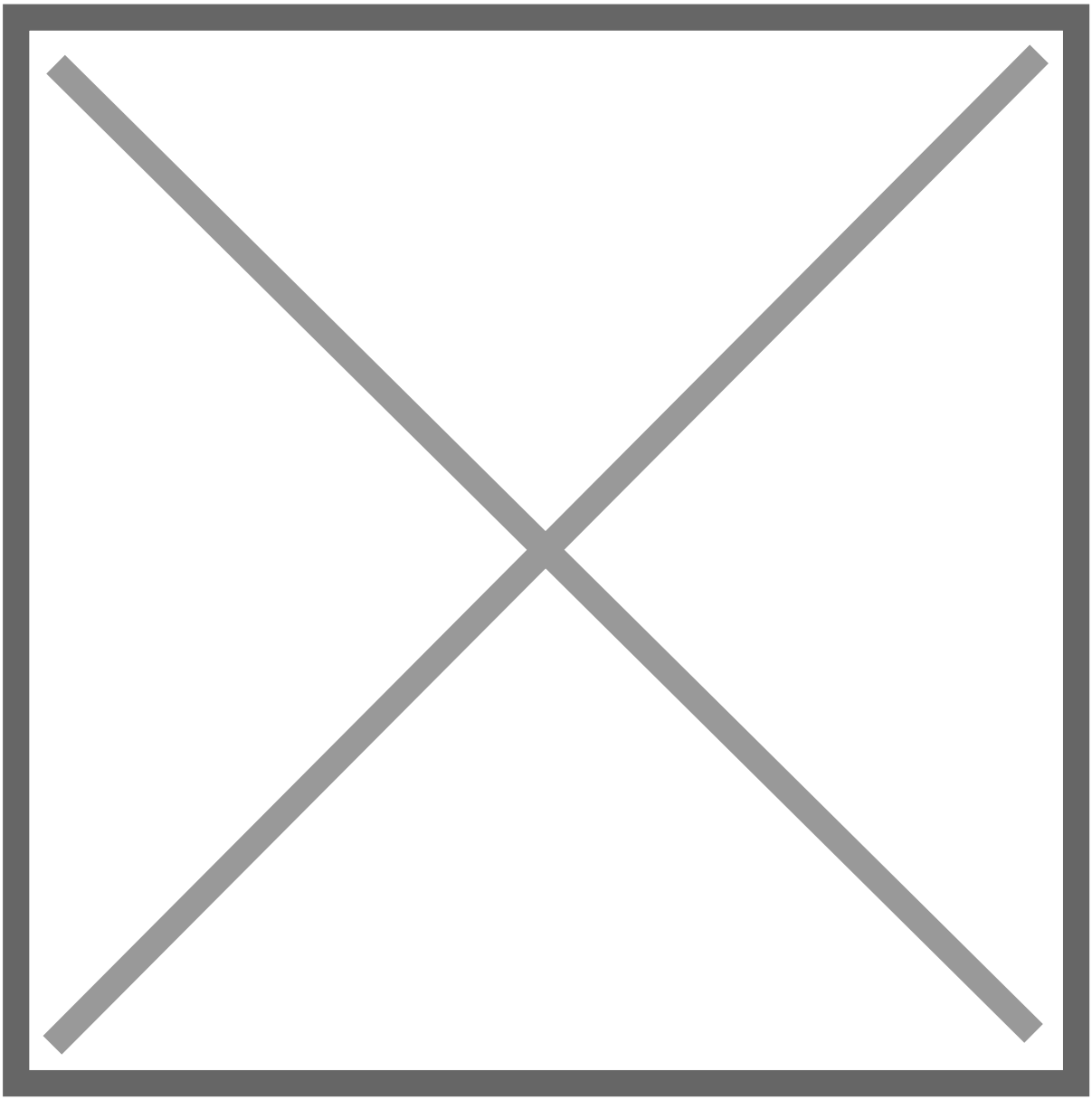
```
https://bookstack.mondomaine.com
```

Compte admin

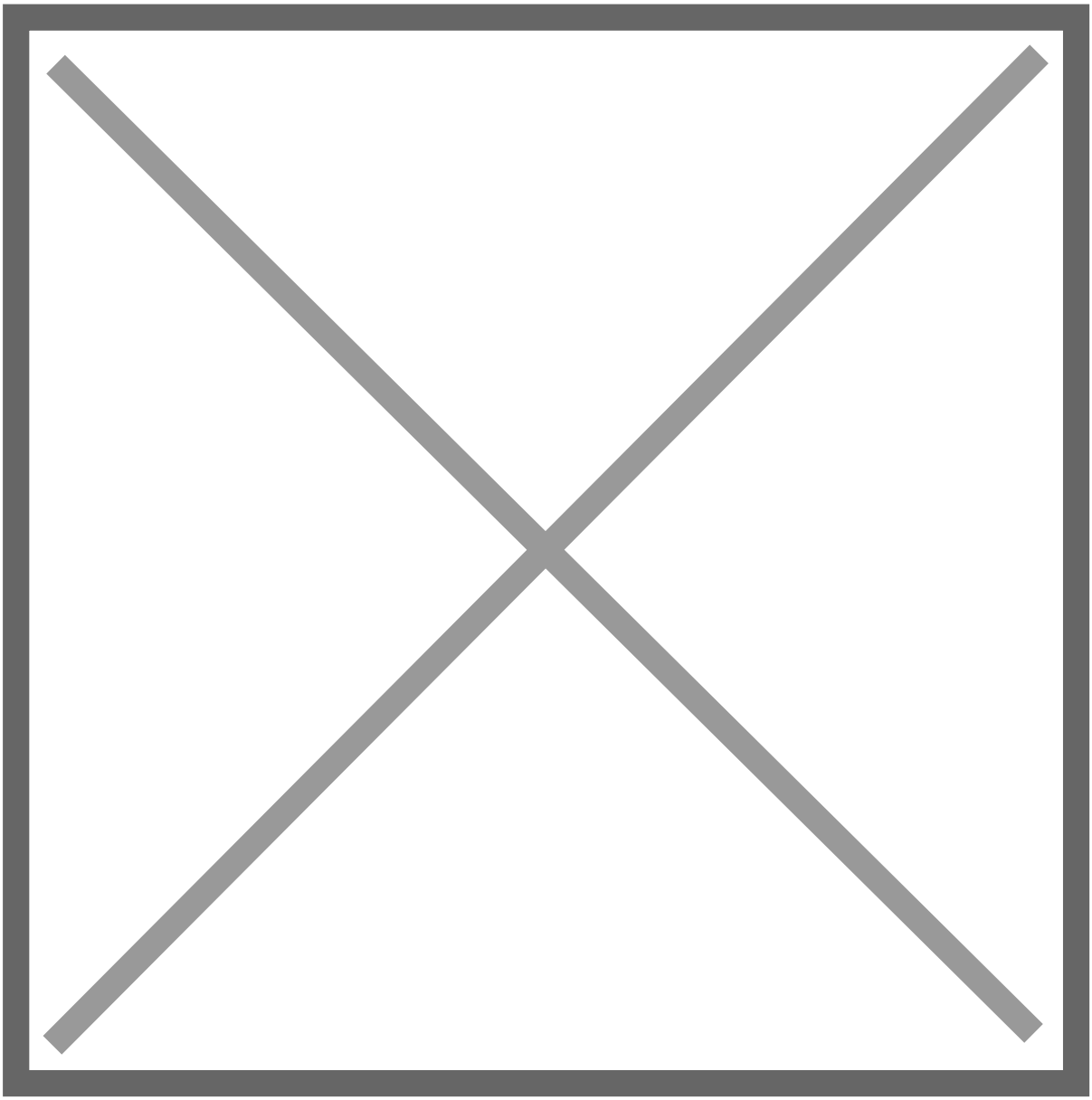
Lors du premier lancement il faudra indiquer le compte administrateur par défaut:

- **E-mail** => admin@admin.com
- **Mot de passe** => password

Validez en cliquant sur **Se Connecter**:

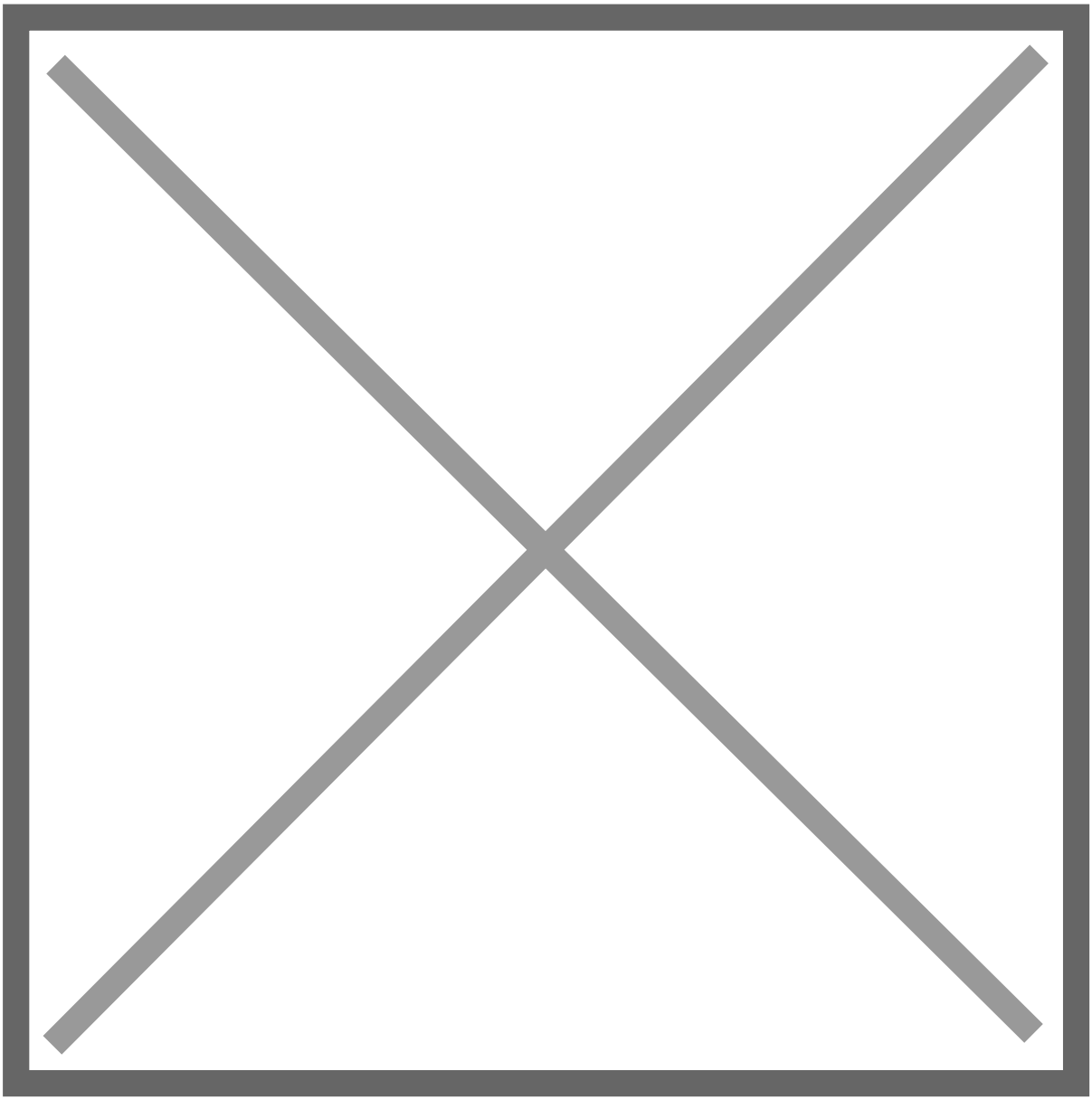


Bienvenue sur la page principale de BookStack:

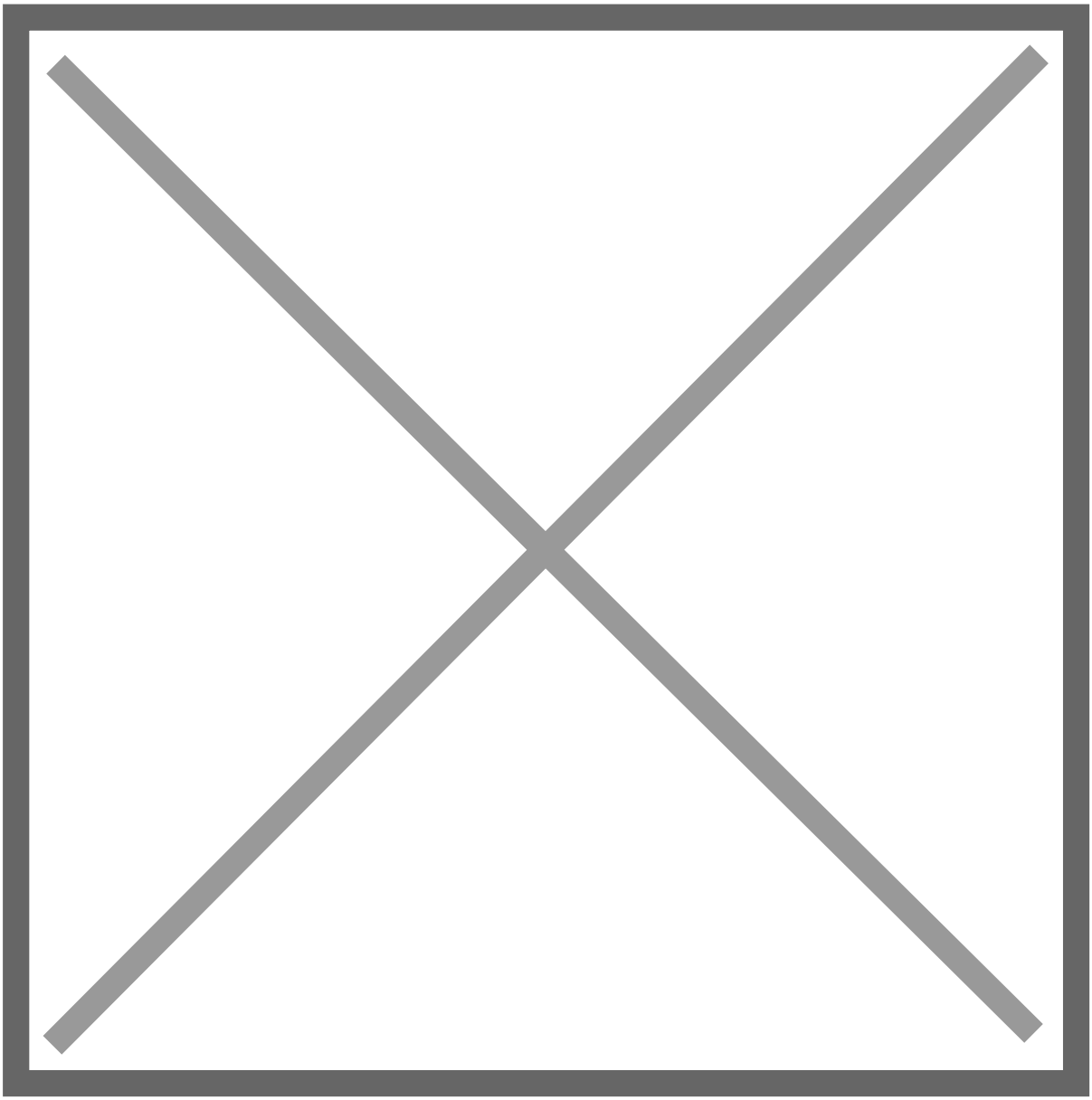


Changer la langue

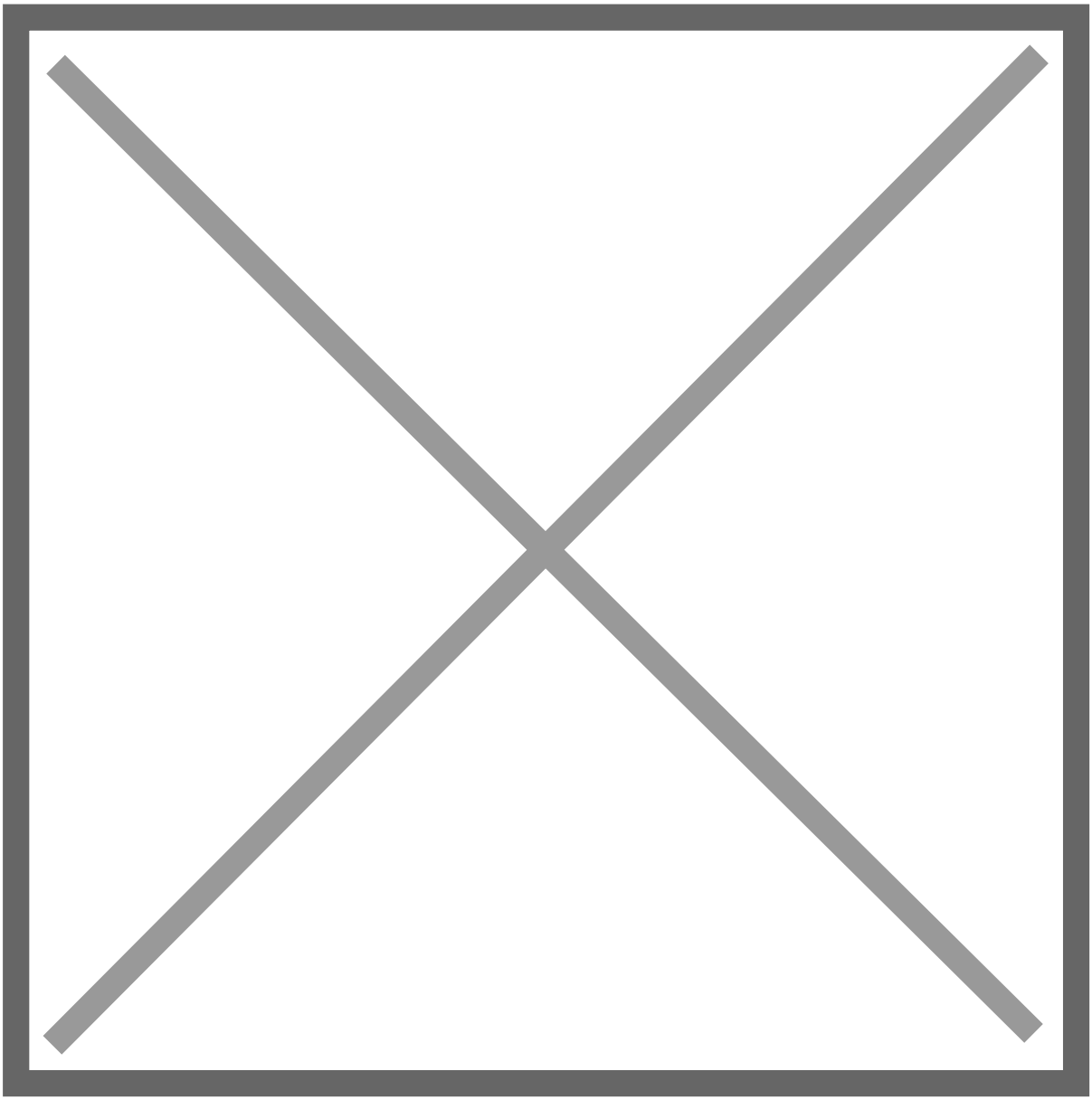
Allez en haut à droite, cliquez sur **Admin**, ensuite sur **My Account**:



Dans **Preferred Language**, sélectionnez la langue de votre choix, ici le français, validez avec **Save**:

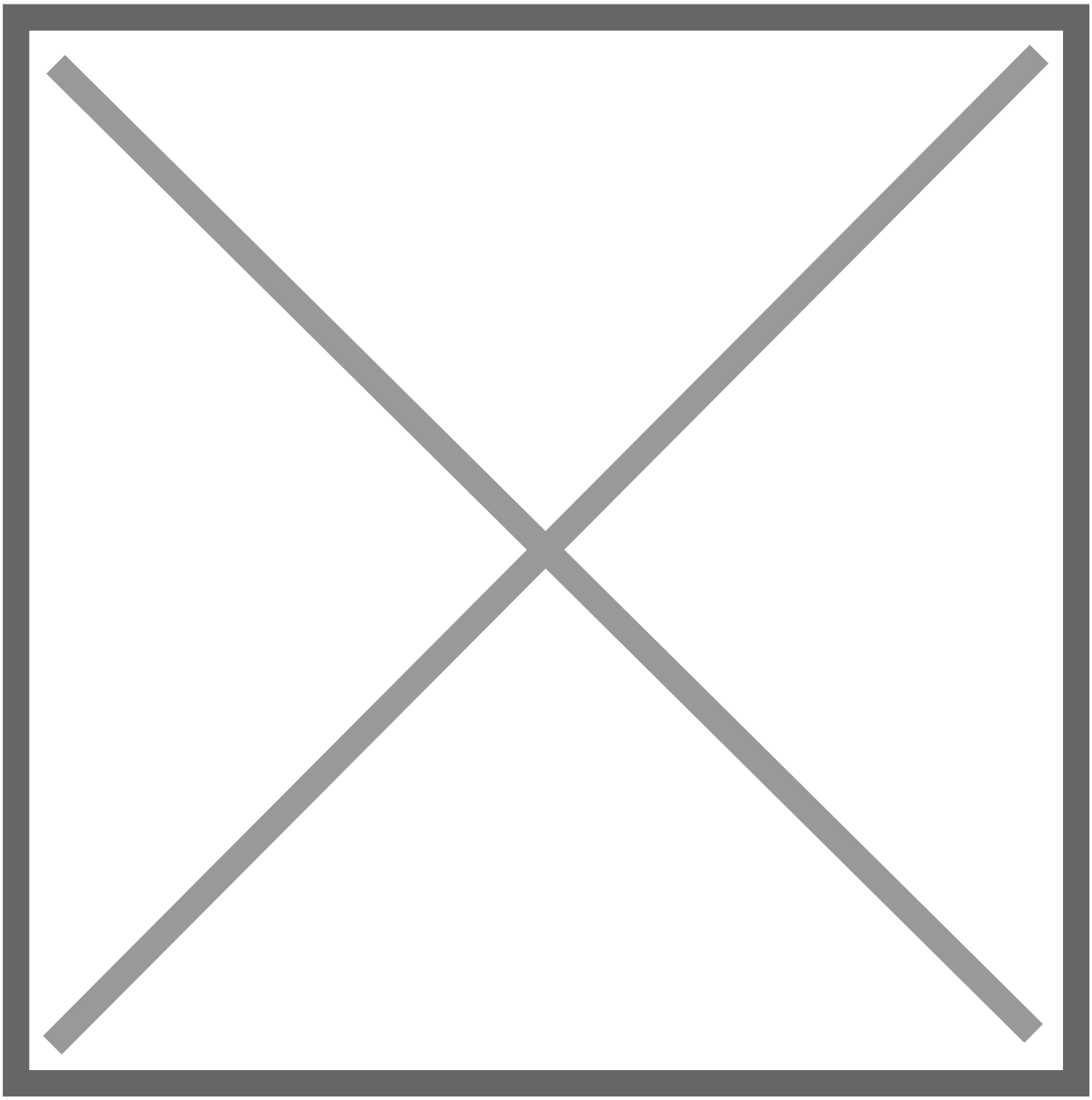


BookStack est désormais en français:

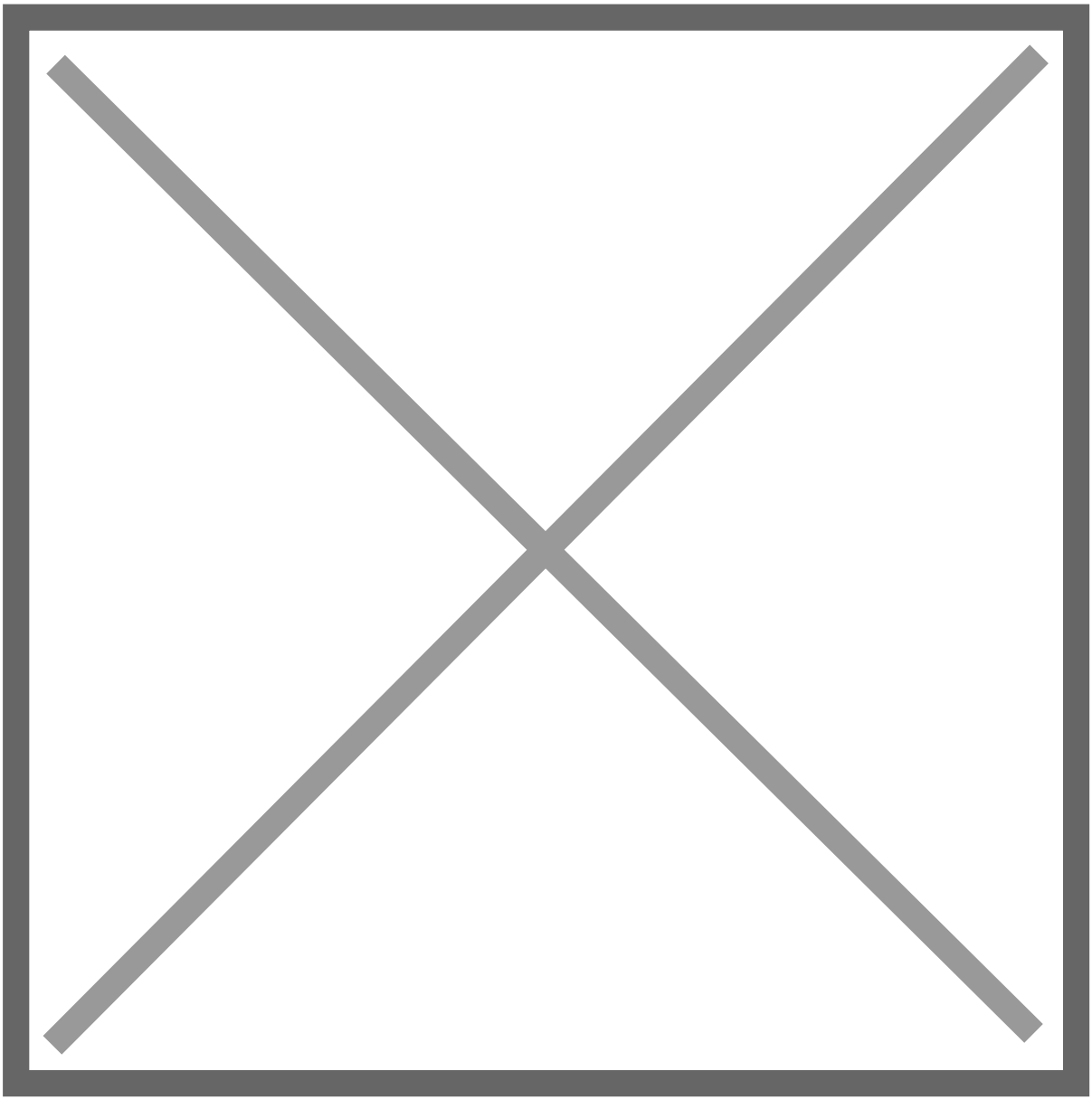


Changer le Nom et l'E-mail du profil

Allez en haut à droite, cliquez sur **Admin**, ensuite sur M**on compte**:

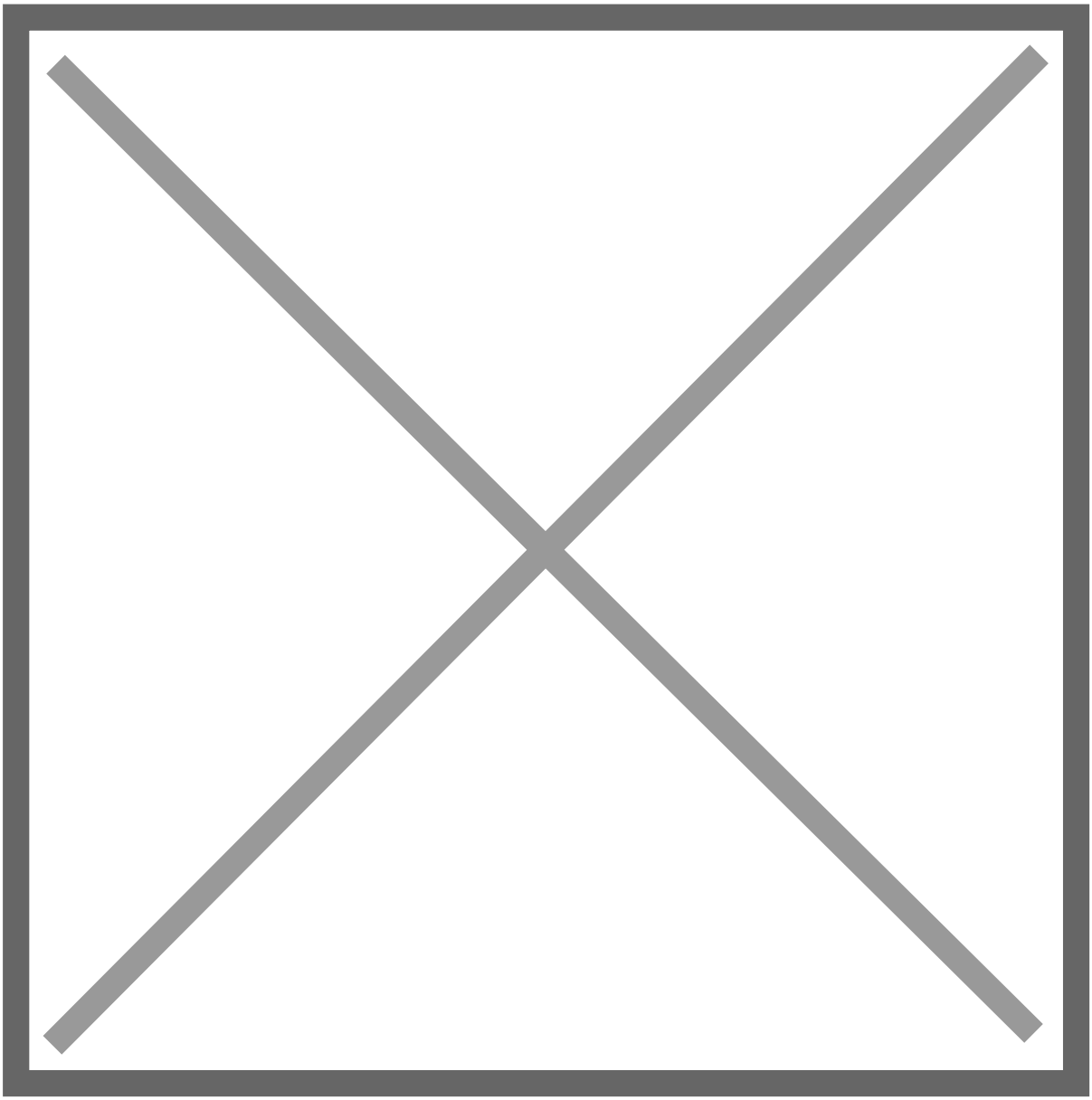


Changez **Nom** et **E-mail** avec les informations souhaitées, validez avec **Enregistrer**:

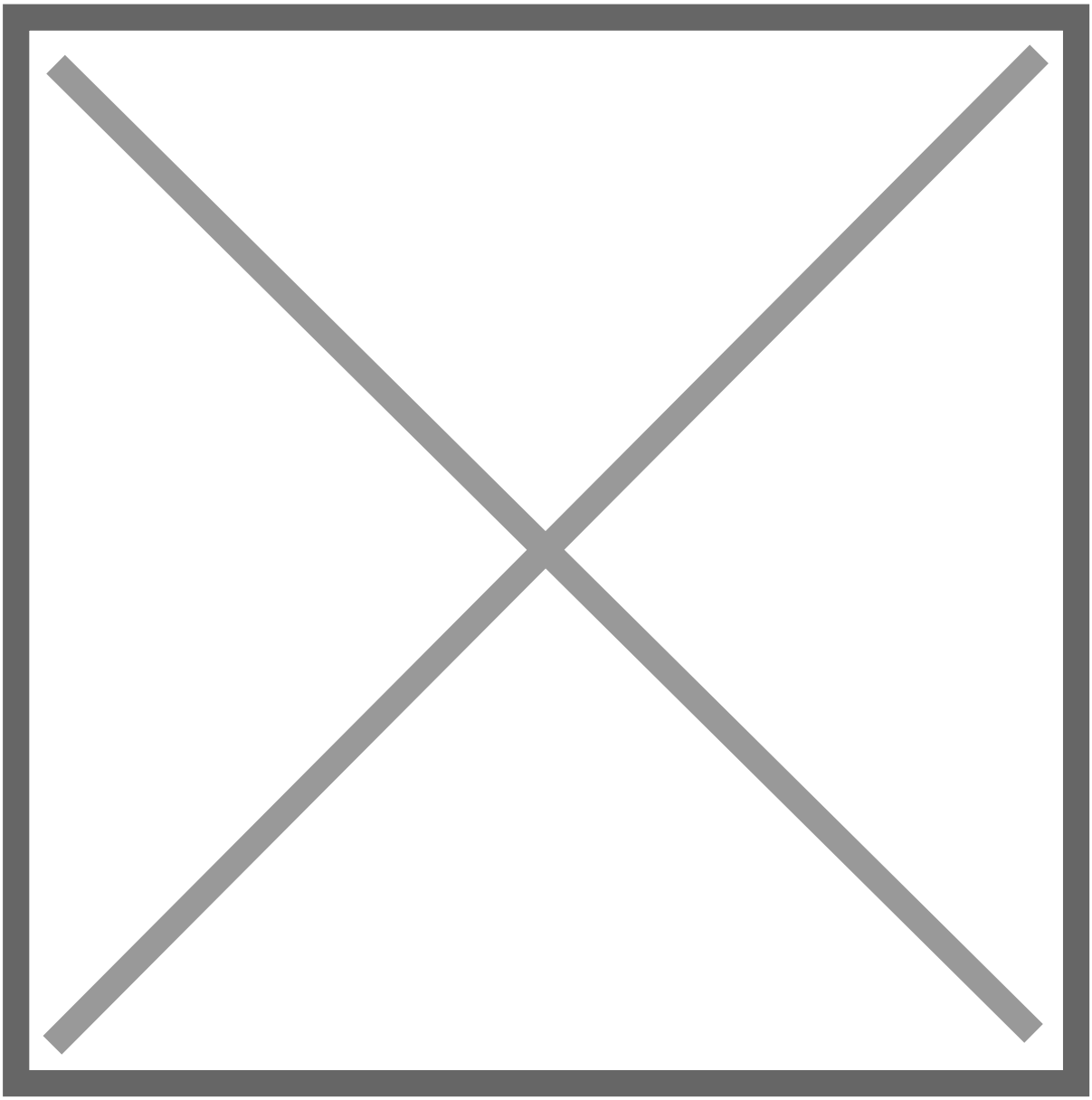


Changer le mot de passe

Allez en haut à droite, cliquez sur **Admin**, ensuite sur M**on compte**:

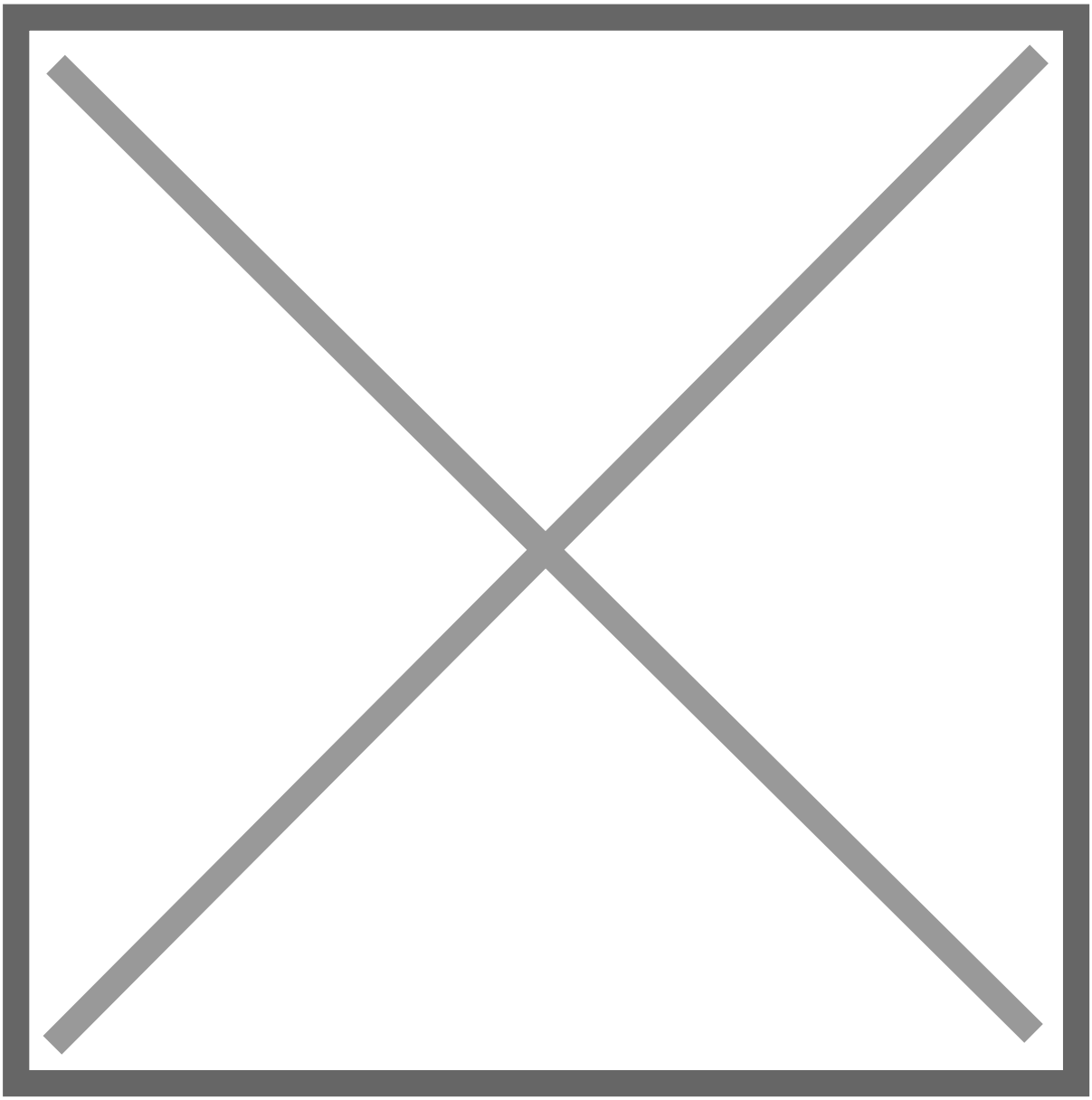


Dans le menu de gauche, sélectionnez **Accès et sécurité**, remplissez les deux champs du mot de passe et validez en cliquant sur **Modifier**:



Activer le mode sombre

Tout simplement sur la page principale:



Bookstack

Backuping and restoring

https://www.bookstackapp.com/docs/admin/backup-restore/

Backup and Restore

While BookStack does not currently have a built-in way to backup and restore content, it can usually be done via the command line with relative ease. The below commands are based on using Ubuntu. If you are using a different operating system you may have to alter these commands to suit.

Backup

There are two types of content you need to backup: Files and database records.

Database

The easiest way to backup the database is via `mysqldump`:

```
1
2
3
4
5
6
7
```

```
# Syntax
## Only specify the `-p` option if the user provided has a password
mysqldump -u {mysql_user} -p {database_name} > {output_file_name}


# Example
mysqldump -u benny bookstack > bookstack.backup.sql
```

If you are using MySQL on Ubuntu, and are using the `root` MySQL user, you will likely have to run the command above with `sudo`:

```
1
```

```
sudo mysqldump -u root bookstack > bookstack.backup.sql
```

The resulting file (`bookstack.backup.sql` in the examples above) will contain all the data from the database you specified. Copy this file to somewhere safe, ideally on a different device.

Files

Below is a list of files and folders containing data you should back up. The paths are shown relative to the root BookStack folder.

- `.env` - File, contains important configuration information.
- `public/uploads` - Folder, contains any uploaded images.
- `storage/uploads` - Folder, contains uploaded page attachments.
- `themes` - Folder, contains any configured visual/logical themes.

Alternatively you could backup up your whole BookStack folder but only the above contain important instance-specific data by default.

The following command will create a compressed archive of the above folders and files:

```
1
```

```
tar -czvf bookstack-files-backup.tar.gz .env public/uploads storage/uploads themes
```

The resulting file (`bookstack-files-backup.tar.gz`) will contain all your file data. Copy this to a safe place, ideally on a different device.

Restore

If you are restoring from scratch follow the installation instructions first to get a new BookStack instance set-up but **do not run the `php artisan migrate` installation step when installing BookStack**. You may need to comment this command out if using an installer script.

If you are using a docker-container-based set-up, restore the database before running the BookStack container. An example of the process using a linuxserver.io-based docker-compose setup can be seen in our video here.

Database

To restore the database you simply need to execute the sql in the output file from the `mysqldump` you performed above. To do this copy your database SQL backup file onto the BookStack or database host machine and run the following:

```
1
2
3
4
5
6
7
8
9
10
```

```
# Syntax
mysql -u {mysql_user} -p {database_name} <
{backup_file_name}
## Only specify the -p if the user provided
has a password

# Example
mysql -u benny -p bookstack <
bookstack.backup.sql

# If using the root user on Ubuntu you may
# have to run the above with root
permissions via sudo:
sudo mysql -u root bookstack <
bookstack.backup.sql
```

If you are restoring to a new version of BookStack you will have to run `php artisan migrate` after restore to perform any required updates to the database.

Files

To restore the files you simply need to copy them from the backup archive back to their original locations. If you created a compressed `bookstack-files-backup.tar.gz` archive as per the backup instructions above you can simply copy that file to your BookStack folder then run the following command:

```
1
```

```
tar -xvzf bookstack-files-backup.tar.gz
```

If you get errors during the above command it may be due to permissions. Change permissions so you can write to the restore locations.

After a backup of the files you should reset the permissions to ensure any write-required locations are writable by the server. The locations required for this can be found in the installation instructions.

Configuration (.env File)

During a restore, you may end up merging various configuration options between your old and new instance `.env` files, to get things working for the new environment. For example, it's common to use the old `.env` settings for most things but use database settings from the `.env` file of a newly created instance.

One thing to be aware of is that you should use the `APP_KEY` value of the old `.env` file since this is used for various features like the encryption of multi-factor authentication credentials. Changing the `APP_KEY` may cause such features to break.

URL Changes

If you are restoring into an environment where BookStack will run on a different URL, there are a couple of things you'll need to do after restoring everything:

- Within the `.env` config file update the `APP_URL` value to exactly match your new base URL.
- Run the "Update System URL" command to update your database content to use your new URL.

If you migrated web-server configuration files, you may also need to tweak those to correctly use the new URL.

Astuces et amélioration

stats page vue :

https://github.com/BookStackApp/BookStack/issues/4775#issuecomment-2474510624

une solution extérieur :

https://www.bookstackapp.com/blog/replacing-ga-and-mailchim7p/

sys commande pour gérer boostack

usage	commande
Entrer dans la base de données du conatainer	sdockexecroot seg_mariadb mariadb --user=<mariadb-username> --password=<mariadb-usr-pwd>

Bookstack

Structure base et dev

Tables :

```
MariaDB [bookstack]> show tables;
+-----------------------+
| Tables_in_bookstack   |
+-----------------------+
| activities            |
| api_tokens            |
| attachments           |
| books                 |
| bookshelves           |
| bookshelves_books     |
| cache                 |
| chapters              |
| comments              |
| deletions             |
| email_confirmations   |
| entity_permissions    |
| failed_jobs           |
| favourites            |
| images                |
| imports               |
| jobs                  |
| joint_permissions     |
| mfa_values            |
| migrations            |
| page_revisions        |
| pages                 |
| password_resets       |
| permission_role       |
| references            |
| role_permissions      |
| role_user             |
| roles                 |
| search_terms          |
| sessions              |
| settings              |
| social_accounts       |
| sort_rules            |
```

| tags |
| user_invites |
| users |
| views |
| watches |
| webhook_tracked_events |
| webhooks |
+-----------------------+

Hacks : notify-page-updates-for-tagged-books

essai 27 bookstack-module.json

dans
nicolas@msi-01:/srv/bookstack$ nano docker-compose .yml

image: linuxserver/bookstack:26.03.20260315

Puis :

dans le .env !

nicolas@msi-01:/srv/bookstack$ sdock compose down && sdock compose up -d
nicolas@msi-01:/srv/bookstack$ sdockexec <container_name> bash

root@597eaf963fbf:/# apk update && apk add --no-cache php85-cli php85-mbstring php85-xml php85-zip

root@597eaf963fbf:/# wget https://www.bookstackapp.com/hack-modules/notify-tagged-page-updates.zip
root@597eaf963fbf:/# mv notify-tagged-page-updates.zip /app/www/
root@597eaf963fbf:/# cd /app/www/ && php artisan bookstack:install-module notify-tagged-page-updates.zip
root@597eaf963fbf:/# apk add nano
root@597eaf963fbf:/# nano /app/www/themes/custom/modules/notify-page-updates-for-tagged-books/functions.php

Modif dans la fonction ;

```
24 public function toMail(User $notifiable): MailMessage
25 {
26 /** @var Page $page */
27 $page = $this->detail;
28 $updater = $this->user;
29
30 return (new MailMessage())
31 // ->subject('BookStack page update notification')
```

```
32 // ->line("La page \"{$page->name}\" has been updated by \"{$updater->name}\"")
33 ->subject('BookStack : notification de mise à jour de page')
34 ->line("The page \"{$page->name}\" a été mise à jour par \"{$updater->name}\"")
35 ->action('View Page', $page->getUrl());
36 }
```

root@597eaf963fbf:/# exit

puis :

nicolas@msi-01:/srv/bookstack$ sdock retart <nom_container>

ATTENTION :

NOM DU TAG : notify en minuscule
Rôles à pousser : casse identique au nom du rôle, plusieurs rôles séparés par une virgule.

Etat courant :

Thème BookStack custom

Modules installés

notify-page-updates-for-tagged-books

- Source : https://www.bookstackapp.com/hacks/notify-tagged-page-updates/
- Installé le : 2026-03-17
- Testé sur : v26.03

Fonctionnement : Envoie un mail aux utilisateurs des rôles listés dans un tag `notify` (séparateur virgule) appliqué au **livre** (pas au chapitre ni à la page).

Exemple de tag sur le livre :

- Nom : `notify`
- Valeur : `mn-et-nf,Admin`

Modifications apportées au functions.php original :

- Textes des mails traduits en français
- Logs de debug ajoutés (niveau debug, silencieux en prod)

nouvelle fonction avec 1 mail tous les xxx

Ce qui a changé par rapport à ton fichier actuel :

- Constante `NOTIFY_COOLDOWN_HOURS = 4` en haut du fichier — **un seul endroit à modifier** pour changer la fréquence
- Le cooldown est vérifié **avant** toute requête SQL — si actif, on sort immédiatement sans toucher la base
- Le cooldown est armé **après** l'envoi réussi des mails — si une erreur survient avant, le prochain enregistrement retente
- Tous les `\Log::info` remplacés par `\Log::debug` — silencieux en prod, visibles si `LOG_LEVEL=debug`
- Message du mail entièrement en français

Écriture d'une nouvelle fonctionnalité : s'abonner aux notification de pages modifiées dans un livre.

Plan d'action — Digest de suivi de livres BookStack

Contexte

Remplacer le hack `notify-tagged-page-updates` (notification par rôle, immédiate) par un système de **suivi individuel par utilisateur** avec **digest périodique**.

Ce qu'on sait déjà

- Pas d'événement `CRON/DAILY` dans `ThemeEvents.php` → le déclencheur du digest sera un **cron sur l'hôte** (`msi-01`)
- Le stockage sera un **fichier JSON externe** monté en volume, jamais touché par les mises à jour de l'image
- L'abonnement se fait via un **tag** `follow:nicolas,marino` posé manuellement sur le livre — choix individuel, pas de groupe

> Pas totalement d'accord avec le cron externe : cela fragilise le système. au moins une cron intégré au conteneur serait plus sécurisé. mais utiliser la fonction schedule de laravel serait le plus robuste.

Les 3 chantiers

Chantier 1 — Le fichier JSON

- Définir son emplacement dans le volume (`/srv/bookstack/bookstack_data/hack-digest.json`)
- Définir sa structure définitive
- Le créer vide au départ

Chantier 2 — Réécriture du `functions.php`

- Écouter `PAGE_UPDATE` → lire le tag `follow:` sur le livre → accumuler dans le JSON
- Remplacer toute la logique d'envoi immédiat par l'accumulation

Chantier 3 — Le digest

- Écrire le script d'envoi (commande artisan custom ou script PHP standalone)
- Formater l'email récapitulatif (liste des pages modifiées par livre)
- Brancher le cron sur l'hôte à l'heure souhaitée

Questions à trancher en début de prochaine session

1. **Fréquence du digest** — quotidien à heure fixe ? Paramétrable par utilisateur ?
2. **Emplacement du JSON** — à confirmer selon la structure de tes volumes
3. **Format de l'email** — simple liste texte ou HTML avec liens cliquables ?

Bonne soirée ! □□

MacBoock Pro

Tip, pb, solution pour ubuntu sur macboock

MacBoock Pro

Redemarrer sur clé usb

Comment démarrer un Mac à partir d'un support USB

Le démarrage de votre Mac à partir d'une clé USB est assez simple.

1. Insérez le support de démarrage USB dans une port USB libre.
2. Appuyez sur le bouton Marche/Arrêt pour allumer votre Mac (ou alors redémarrez votre Mac s'il est déjà allumé).
3. Lorsque vous entendez le bip de démarrage, appuyez sur la touche Option et maintenez-la enfoncée. Le fait de maintenir cette touche enfoncée vous donne accès au gestionnaire de démarrage d'OS X. Une fois que l'écran du gestionnaire de démarrage apparaît, relâchez la touche Option. L'utilitaire recherchera tous les lecteurs disponibles qui contiennent du contenu amorçable.
4. À l'aide du curseur de votre souris ou des touches fléchées du clavier, sélectionnez la clé USB à partir de laquelle vous souhaitez démarrer.
5. Une fois sélectionné, appuyez sur la touche Retour ou double-cliquez sur votre sélection. La machine va alors démarrer à partir de la clé USB.

REMARQUE : plusieurs périphériques USB sont connectés à votre Mac ? Ne vous inquiétez pas. Le gestionnaire de démarrage ne liste que les lecteurs qui comportent un contenu amorçable.

proxy public proxy interne

D'après votre description, vous avez une architecture avec :
1. Un serveur frontal Nginx qui fait office de reverse proxy public
2. Un serveur LAN qui héberge des applications Docker
3. Les applications sont accessibles en interne via `nommachine.local:port_container`

Si vous rencontrez des erreurs lors de l'accès depuis Internet, voici mes recommandations :

Solution recommandée : Oui, installez un second Nginx sur le serveur d'applications

Pourquoi :
1. **Meilleure gestion du routage** : Le proxy frontal devrait parler à un seul endpoint propre (le nouveau Nginx) plutôt qu'à divers ports de containers
2. **Centralisation de la config** : Vous pourrez gérer toutes les règles de routage LAN au même endroit
3. **Résolution des erreurs** : Beaucoup de problèmes viennent des headers HTTP mal propagés (Host, X-Forwarded-For, etc.)

Configuration suggérée :
```
[Internet]
→ Nginx frontal (public IP)
→ Nginx interne (sur le serveur Docker, écoute sur un port fixe)
→ Containers (sur leurs ports respectifs)
```

Avantages :
- Vous pouvez configurer proprement les headers HTTP
- Meilleure gestion du SSL (terminaison au proxy frontal ou interne selon votre besoin)
- Plus simple à maintenir (une seule entrée/sortie sur le serveur Docker)
- Possibilité d'ajouter des règles de routage complexes côté LAN

Alternative :
Vous pourriez tout configurer sur le Nginx frontal, mais cela devient vite complexe si vous avez beaucoup de services et moins sécurisé (exposition des ports internes).

Les erreurs que vous voyez sont probablement dues soit :
1. À des problèmes de résolution DNS interne/externe
2. À des headers HTTP mal propagés
3. À des timeouts entre le proxy frontal et les containers

Une configuration à deux niveaux de proxy est une pratique courante et recommandée dans ce type d'architecture.

La réponse dépend de votre architecture et de vos besoins en sécurité, mais voici les meilleures pratiques :

1. **Solution la plus simple et recommandée (Terminaison SSL unique)**
  - **Seul le proxy frontal gère le SSL**
    - Le trafic entre les deux Nginx (frontal et interne) reste en **HTTP** (LAN)
    - Configuration :
      ```nginx
      # Sur le proxy frontal (public)
      server {
          listen 443 ssl;
          server_name app.votre-domaine.com;
          ssl_certificate /chemin/vers/cert.pem;
          ssl_certificate_key /chemin/vers/key.pem;

          location / {
              proxy_pass http://nommachine.local:80;  # Communication interne non chiffrée
              proxy_set_header Host $host;
              proxy_set_header X-Real-IP $remote_addr;
          }
      }
      ```
    - **Avantages** :
      - Pas besoin de Certbot sur le serveur interne
      - Moins de maintenance (un seul certificat à renouveler)
      - Performances accrues (pas de double chiffrement/déchiffrement)

  - **Quand l'utiliser ?**
    - Si votre LAN est sécurisé (réseau privé fiable, pas besoin de SSL interne).

---

2. **Solution plus sécurisée (SSL end-to-end)**
  - **Proxy frontal ET interne ont leur propre SSL**
    - Le frontal fait une terminaison SSL, puis **reverse proxy vers le second Nginx en HTTPS**
    - Configuration :
      ```nginx
      # Sur le proxy frontal
      location / {
          proxy_pass https://nommachine.local:443;  # Communication interne chiffrée
          proxy_ssl_verify off;  # Désactive la vérification du certificat (optionnel)
      }

      # Sur le proxy interne
      server {
          listen 443 ssl;
      ```

```
        server_name nommachine.local;
        ssl_certificate /chemin/vers/cert-interne.pem;
        ssl_certificate_key /chemin/vers/key-interne.pem;

        location / {
            proxy_pass http://conteneur:port;  # Vers le container Docker
        }
    }
    ```
  - **Avantages** :
    - Chiffrement même sur le LAN (utile si le réseau n'est pas 100% sécurisé)
    - Meilleure conformité pour certains standards (HIPAA, PCI DSS)

- **Quand l'utiliser ?**
  - Si votre LAN n'est pas totalement sécurisé.
  - Si vous avez des données ultra-sensibles (médical, bancaire, etc.).

---

3. **Alternative hybride (SSL entre proxies + HTTP vers les containers)**
- **Proxy frontal en HTTPS → Proxy interne en HTTPS → Containers en HTTP**
  - Moins de charge que le SSL end-to-end, mais sécurise le trafic inter-serveurs.

Faut-il utiliser Certbot sur le second serveur ?
- **Oui** si :
  - Vous voulez une sécurité renforcée (HTTPS entre proxies).
  - Le nom de domaine interne (`nommachine.local`) est accessible publiquement (peu probable).
- **Non** si :
  - Le trafic interne est en HTTP (solution 1).
  - Vous utilisez un certificat auto-signé pour la communication interne.

Recommandation finale
Dans **90% des cas**, la **Solution 1 (terminaison SSL unique)** suffit, car :
- Le LAN est considéré comme un environnement de confiance.
- Évite la complexité de gestion de plusieurs certificats.

Si vous optez pour HTTPS interne, un certificat auto-signé peut suffire (pas besoin de Certbot).
Utilisez Certbot uniquement si le second Nginx doit servir un domaine public.

□□ **Vérifiez aussi** :
- Que les headers (`Host`, `X-Forwarded-For`, `X-Forwarded-Proto`) sont bien transmis.
- Que les timeouts (`proxy_read_timeout`, `proxy_connect_timeout`) sont adaptés.
- Que le DNS interne (`nommachine.local`) fonctionne correctement entre les serveurs.

Si les erreurs persistent, examinez les logs (`sudo tail -f /var/log/nginx/error.log`) des deux proxies.

Pb réseau DELL-9010

La résolution avec claude :

https://claude.ai/share/958e0e47-07a1-46f7-9dc8-56dcce506ff4

Mattermost

Sympa (pas sympa du tout)

□□ Méthode de Réinstallation "Ultra Propre" MariaDB
L'objectif est d'effacer tous les fichiers de configuration, les utilisateurs et les bases de données MariaDB pour garantir que la prochaine installation est vierge.

Étape 1 : Purge et Suppression Complète
Cette étape supprime tous les paquets et tous les fichiers de configuration associés.

Arrêtez MariaDB :

Bash

sudo systemctl stop mariadb
Désinstallez et purgez MariaDB : L'option purge est essentielle car elle supprime les fichiers de configuration qui persistent après une simple remove.

Bash

sudo apt purge mariadb-server mariadb-client -y
Supprimez les dépendances inutiles :

Bash

sudo apt autoremove -y
Supprimez les fichiers de données et de configuration résiduels : Ceci efface toutes les bases de données existantes (y compris les reliquats du socket, des logs, et de la DB Sympa).

```
Bash
 sudo rm -rf /etc/mysql /var/lib/mysql /var/log/mysql /run/mysqld
```

Attention : Cette commande supprime définitivement toutes les bases de données MySQL/MariaDB de cette machine. Confirmez que vous n'avez pas d'autres données importantes à conserver.

Étape 2 : Réinstallation et Sécurisation
Réinstallez les paquets MariaDB :

Bash

```
sudo apt update
sudo apt install mariadb-server mariadb-client -y
```

Vérifiez le statut : MariaDB devrait démarrer automatiquement.

Bash

```
sudo systemctl status mariadb
```

Sécurisez l'installation (méthode manuelle) : Puisque mariadb_secure_installation ne fonctionne pas et que le root TCP/IP est souvent bloqué, nous devons débloquer le root via le socket Unix :

Bash

```
# 1. Connectez-vous via le socket (qui est maintenant autorisé pour root du système)
sudo mariadb
```

```
# 2. Définissez le mot de passe root pour l'accès TCP/IP (dans la console MariaDB>)
ALTER USER 'root'@'localhost' IDENTIFIED BY 'VOTRE_MOT_DE_PASSE_ROOT';
FLUSH PRIVILEGES;
\q
```

Étape 3 : Création de la DB Sympa
Vous pouvez maintenant créer la DB Sympa sur une base MariaDB saine.

Connectez-vous à MariaDB via TCP/IP :

Bash

```
mariadb -u root -h 127.0.0.1 -p
##(Utilisez VOTRE_MOT_DE_PASSE_ROOT)
```

Créez l'utilisateur et la base de données Sympa :

```
##SQL
CREATE DATABASE sympa CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'sympa'@'localhost' IDENTIFIED BY 'VOTRE_MOT_DE_PASSE_SYMPA';
GRANT ALL PRIVILEGES ON sympa.* TO 'sympa'@'localhost';
FLUSH PRIVILEGES;
\q
```

==========

?? Action : Modification du fichier `50-client.cnf`

Ajoutez la ligne `socket = /run/mysqld/mysqld.sock` sous la section `[client]`.

1. **Ouvrez le fichier :**
   Bash
   ```
   sudo nano /etc/mysql/mariadb.conf.d/50-client.cnf
   ```
2. **Ajoutez la ligne critique :**
   Ini, TOML
   ```
   # Fichier 50-client.cnf

   # This group is read by the client library
   # Use it for options that affect all clients, but not the server
   #

   [client]
   # Ajout de la ligne pour forcer le chemin du socket
   socket = /run/mysqld/mysqld.sock

   # ... le reste du contenu natif
   ```

Pb avec VS code

> nicolas@asrock:~$ GDK_BACKEND=x11 code
> nicolas@asrock:~$ GDK_BACKEND=wayland code
> nicolas@asrock:~$ nano /home/nicolas/.vscode/argv.json
> nicolas@asrock:~$ GDK_BACKEND=x11 code
> nicolas@asrock:~$ setxkbmap -query
> WARNING: Running setxkbmap against an Xwayland server
> rules: evdev
> model: pc105
> layout: us
> options: kpdl:comma
> nicolas@asrock:~$ setxkbmap fr
> WARNING: Running setxkbmap against an Xwayland server
> **nicolas@asrock:~$ echo $XDG_SESSION_TYPE**
> **wayland**
> **nicolas@asrock:~$ gsettings set org.gnome.desktop.input-sources sources "[('xkb', 'fr')]"**
> **nicolas@asrock:~$ code**
> **nicolas@asrock:~$ dconf write /org/gnome/desktop/input-sources/sources "[('xkb', 'fr')]"**
> **nicolas@asrock:~$**

Python

Python

PyLogo - discussion sur le projet

Excellente question théorique ! Oui, absolument, et c'est même très intéressant pédagogiquement.

Approches possibles pour les objets en Logo

1. **Approche "Dictionnaire/Table" (la plus simple)**

Étendre la syntaxe actuelle avec des structures de données :

```
(* Nouveau type : table associative *)
table = "{" [ paire { "," paire } ] "}" ;
paire = identificateur ":" expression ;

(* Accès aux propriétés *)
acces_propriete = variable "." identificateur ;

(* Exemple d'utilisation *)
DONNE :vetement {nom: "Chemise", couleur: "Bleu", taille: 42}
ECRIS :vetement.nom
```

Avantages :

- Naturel pour les enfants (proche des objets du monde réel)
- Pas de concept de "classe" à enseigner
- Extension minimale de la grammaire

2. **Approche "Prototype" (style JavaScript)**

Pas de classes, seulement des objets qui se clonent :

```
(* Création d'un prototype *)
PROTOTYPE identificateur [ liste_proprietes ]
  { instruction }
FIN

(* Exemple *)
PROTOTYPE Vetement :nom :couleur
  DONNE :this.nom :nom
  DONNE :this.couleur :couleur
FIN

NOUVEAU :chemise Vetement "Chemise" "Bleu"
```

Avantages :

- Plus flexible
- Concept de "copie" plus intuitif qu'héritage
- Pas de hiérarchie complexe

3. **Approche "Modèle" (ce que vous suggérez)**

Une forme hybride avec des templates :

```
(* Définition d'un modèle *)
MODELE identificateur
  { declaration_propriete }
FIN

declaration_propriete = identificateur [ "=" expression ] ;

(* Instanciation *)
NOUVEAU :instance identificateur
DONNE :instance.propriete valeur
```

Avantages :

- Vocabulaire adapté aux enfants ("modèle" vs "classe")
- Séparation claire structure/données
- Proche de votre proposition

4. **Approche "Liste nommée" (ultra-simple)**

Réutiliser les listes existantes avec des conventions :

```
; Pas de nouvelle syntaxe !
DONNE :vetement [nom "Chemise" couleur "Bleu" taille 42]

; Fonction utilitaire
POUR OBTENIR :objet :cle
  ; Cherche la clé dans la liste
FIN

ECRIS OBTENIR :vetement "nom"
```

Avantages :

- Zéro changement grammatical
- Enseigne la manipulation de listes
- Très flexible

Ma recommandation pour un contexte éducatif

Je pencherais pour **l'Approche 4 (Listes nommées) au départ**, puis **Approche 1 (Dictionnaires)** si besoin :

Pourquoi ?

1. **Progressivité** : Les enfants maîtrisent d'abord les listes, puis ajoutent la sémantique d'objet
2. **Pas de magie** : Tout est visible (c'est juste des listes !)
3. **Découverte** : "Tiens, si on mettait les noms et valeurs ensemble..."
4. **Transition naturelle** vers les vrais objets plus tard

Extensions de la grammaire EBNF

Si on voulait ajouter des tables/objets :

```
(* Structures de données *)
facteur = nombre
        | variable
        | chaine
        | table          (* NOUVEAU *)
        | liste
        | ... ;

table = "{" [ entree_table { "," entree_table } ] "}" ;
entree_table = identificateur ":" expression ;

(* Accès *)
acces_membre = variable "." identificateur
             | variable "[" expression "]" ;  (* accès par clé *)

(* Modification *)
affectation_membre = variable "." identificateur expression
                   | variable "[" expression "]" expression ;
```

Question philosophique

Faut-il ajouter les objets à Logo ?

Pour :

- Modéliser le monde réel
- Préparer aux langages modernes
- Organiser le code complexe

Contre :

- Logo enseigne la **pensée procédurale** d'abord
- Risque de complexifier trop tôt
- Les listes suffisent souvent

Mon avis : Commencer sans objets, les introduire comme "bibliothèques de fonctions" qui manipulent des listes structurées, puis éventuellement formaliser la syntaxe si le besoin émerge naturellement.

C'est cohérent avec la philosophie Logo : découverte progressive, pas abstraction imposée ! □□

Python

Next to do

Excellent plan pour la prochaine session !

Récapitulatif de ce qu'on a aujourd'hui

Grammaire Logo sans ambiguïté

- Variables : `:nom`
- Chaînes : `"texte"`
- Appels de fonction : `nom args`
- Priorités documentées dans l'EBNF

Interpréteur fonctionnel

- Procédures avec paramètres
- Récursion
- SI/SINON/IFELSE
- REPETE
- Support Turtle graphique (`--turtle`)
- Messages d'erreur adaptés aux enfants

Pédagogie

- Messages bienveillants
- Suggestions de correction
- Pas de traceback Python

Plan pour la prochaine session

1. Refactoring du code

```
pylogo/
├── lexer.py          # Analyse lexicale
├── parser.py         # Analyse syntaxique
├── ast_nodes.py      # Nœuds AST
```

```
├── interpreter.py    # Exécution
├── turtle_backend.py # Backend graphique
├── commands.py       # Commandes Logo
└── main.py          # Point d'entrée
```

2. Primitives manquantes

- Listes : `PREMIER`, `DERNIER`, `ELEMENT`
- Mathématiques : `ABS`, `MIN`, `MAX`, `PUISSANCE`
- Chaînes : `CONCAT`, `LONGUEUR`
- Fichiers : `LIS`, `ECRIS_FICHIER`

3. Interface enfant

Option Desktop : Tkinter/PyQt **Option Web** : HTML + Ace Editor + Python backend

Avec :

- Zone code (éditeur)
- Zone graphique (turtle)
- Console
- Boutons : ▶ Exécuter | □□ Sauvegarder | □□ Charger | □□ Nouveau

À avoir pour la prochaine fois

- Le code actuel (`PyLogo_NoAmbiguity.py`)
- La grammaire EBNF
- Les décisions sur :
  - Desktop vs Web ?
  - Quelles primitives prioriser ?
  - Style graphique souhaité ?

Bravo pour ce travail aujourd'hui ! On a construit un Logo propre, sans ambiguïté, avec une vraie pédagogie. C'est une excellente base pour la suite ! □□□

À la prochaine session ! □□

debeuggage lionel

Nicolas, voici une analyse structurée de votre problème et les pistes à explorer, en partant du symptôme le plus révélateur : **le ping vers 8.8.8.8 échoue avec "Défaillance générale"**, alors que le ping vers la box Orange fonctionne.

Diagnostic du problème

Symptômes clés

- ✅ **Connexion locale OK** : Ping vers la box (ex: 192.168.1.1) fonctionne → Le PC communique avec le réseau local.
- ❌ **Accès Internet bloqué** : Ping vers 8.8.8.8 (DNS Google) échoue → Problème de **connectivité sortante** vers Internet.
- ❌ **Navigateurs bloqués** : Aucun site accessible → Confirmation du blocage sortante.
- ✅ **Autres appareils OK** : Le problème est **isolé à ce PC**.

Origines probables

1. **Problème de passerelle (gateway) ou de route par défaut**
   - Vérifiez que la **passerelle par défaut** (généralement l'IP de la box, ex: 192.168.1.1) est correctement configurée dans les paramètres IPv4 de la carte Wi-Fi.
   - Commande pour vérifier :

     → Cherchez la ligne **"Passerelle par défaut"** et comparez avec l'IP de votre box.
2. **Blocage par le pare-feu Windows ou un antivirus**
   - Un pare-feu (Windows Defender, antivirus tiers comme Avast, McAfee, etc.) peut bloquer les connexions sortantes.
   - **Test** : Désactivez temporairement le pare-feu et l'antivirus, puis retestez le ping vers 8.8.8.8.
     - Pour désactiver le pare-feu Windows :

       (Réactivez-le après le test avec `netsh advfirewall set allprofiles state on`).
3. **Problème de pilote de la carte Wi-Fi**
   - Un pilote corrompu ou obsolète peut causer des défaillances intermittentes.
   - **Solution** :
     - Ouvrez le **Gestionnaire de périphériques** (Win + X > Gestionnaire de périphériques).

    - Développez **"Cartes réseau"**, faites un clic droit sur votre carte Wi-Fi > **Mettre à jour le pilote**.
    - Si le problème persiste, désinstallez le pilote (clic droit > Désinstaller), puis redémarrez le PC pour forcer la réinstallation automatique.
4. **Paramètres IPv6 conflictuels**
    - Certains routeurs Orange ont des problèmes avec IPv6.
    - **Test** : Désactivez IPv6 sur votre carte Wi-Fi :
        - Allez dans **Paramètres > Réseau et Internet > Wi-Fi > Modifier les options de l'adaptateur**.
        - Clic droit sur votre connexion Wi-Fi > **Propriétés** > Décochez **"Protocole Internet version 6 (TCP/IPv6)"** > OK.
        - Redémarrez le PC et testez.
5. **Problème de MTU (Maximum Transmission Unit)**
    - Un MTU trop élevé peut causer des fragmentations de paquets.
    - **Test** : Réduisez le MTU à 1472 (valeur courante pour éviter les problèmes) :
        - Ouvrez une invite de commandes **en admin** et exécutez :

          (Remplacez `<ID>` par l'ID de votre interface Wi-Fi, visible via `netsh interface ipv4 show subinterfaces`).
6. **Conflit d'IP ou cache réseau corrompu**
    - **Réinitialisez la pile TCP/IP** :

      Puis redémarrez le PC.

7. **Problème matériel (carte Wi-Fi défectueuse)**
    - Testez avec un **câble Ethernet** pour isoler le problème :
        - Si ça fonctionne en Ethernet → Problème lié au Wi-Fi (pilote, carte, ou interférences).
        - Si ça ne fonctionne pas → Problème système (pare-feu, configuration réseau, etc.).

Étapes de dépannage prioritaires

1. Vérifiez la passerelle et les routes

- Exécutez :

  → Vérifiez qu'une route par défaut (`0.0.0.0`) pointe vers l'IP de votre box.
- Si la route manque, réinitialisez-la :

  (Remplacez `<IP_BOX>` par l'IP de votre box, ex: 192.168.1.1).

2. Testez avec un autre DNS

- Dans les paramètres IPv4 de votre carte Wi-Fi, essayez de forcer les DNS de Google :
  - **DNS préféré** : `8.8.8.8`
  - **DNS alternatif** : `8.8.4.4`
  - Validez et testez.

3. Mode sans échec avec réseau

- Redémarrez le PC en **mode sans échec avec réseau** (appuyez sur F8 au démarrage ou via `msconfig`).
  - Si ça fonctionne en mode sans échec → Un logiciel tiers (antivirus, VPN, etc.) bloque la connexion.

4. Réinitialisation complète du réseau

- Exécutez ces commandes **en admin** :

  Puis redémarrez.

Résumé des causes les plus probables

Cause	Test	Solution
Passerelle manquante	`ipconfig /all`	Configurer manuellement la passerelle
Pare-feu/antivirus	Désactiver temporairement	Ajouter une exception ou désinstaller
Pilote Wi-Fi	Mettre à jour/désinstaller	Réinstaller le pilote
IPv6	Désactiver IPv6	Décochez IPv6 dans les propriétés de la carte
MTU trop élevé	`ping 8.8.8.8 -f -l 1472`	Réduire le MTU à 1472
Cache réseau corrompu	`netsh int ip reset`	Réinitialiser la pile TCP/IP

Prochaine étape

Nicolas, pour affiner le diagnostic, pourriez-vous me dire :

1. **Quelle est l’IP de votre box** (visible via `ipconfig /all`) ?
2. **Avez-vous un antivirus ou un VPN installé** sur ce PC ?
3. **Le problème persiste-t-il en mode sans échec avec réseau** ?

Cela nous permettra d’isoler la cause entre **configuration réseau**, **logiciel tiers**, ou **matériel**.