Synology

- Obtenir l’UID et le GID d’un utilisateur
- Comment réinitialiser mon Synology NAS
- Container Manager
  - Homer : Un tableau de bord personnalisable
  - NAS Synology : comment sauvegarder et restaurer un container Docker ?
  - FreshRSS : Agrégateur de flux RSS
  - CyberChef : boite à outils cybersécurité
  - Prise en main de Container Manager
  - DocuSeal : la solution de signature électronique
  - iVentoy : serveur PXE
  - Stirling PDF : La boite à outils PDF
  - NetAlertX : surveillance du réseau
  - Uptime Kuma : Surveiller vos sites web et conteneurs Docker
  - Vaultwarden : Gestionnaire de mot de passe

Obtenir l’UID et le GID d’un utilisateur

Dans ce tutoriel, nous allons apprendre à récupérer l'UID et le GID d'un utilisateur sur un NAS Synology. Vous allez me dire : pourquoi faire ? Et bien, sachez que ceci est utile lorsque l'on utilise un NAS Synology pour exécuter des containers Docker et que l'on souhaite faire tourner un container avec un compte utilisateur spécifique.

Rappels sur les notions de UID et GID

Avant de vous expliquer comment récupérer l'UID et le GID sur un NAS, voyons déjà à quoi correspondent ces deux valeurs.

L'**UID** pour **User Identifier** est un **numéro unique associé à chaque utilisateur** d'un système Linux. Il permet d'**identifier l'utilisateur sans utiliser le nom** et ne peut pas être modifié. Cette information est stockée dans le fichier "**/etc/passwd**".

Le **GID** pour **Group Identifier** est un **numéro unique associé à chaque groupe** d'un système Linux. Le GID permet d'**identifier un groupe sans utiliser le nom** et ne peut pas être modifié. Cette information est stockée dans le fichier "**/etc/group**".

Si un utilisateur ou un groupe est supprimé puis recréé, il n'aura pas le même UID / GID, car ce numéro est incrémenté à chaque fois.

> **Remarque** : le compte super-utilisateur "root" hérite toujours de l'UID "0" et du GID "0".

Récupérer l'UID et le GID d'un utilisateur

Pour récupérer l'UID et le GID d'un utilisateur, nous devons utiliser la ligne de commandes SSH.

La première étape consiste à se rendre dans "**Panneau de configuration**", puis "**Terminal & SNMP**" afin de cocher l'option "**Activer le service SSH**". Ensuite, validez, et à la fin de l'opération, vous pouvez décocher cette option pour éviter d'exposer ce service inutilement si vous n'en avez pas l'usage.

Désormais, nous devons nous connecter en SSH à notre NAS. Vous pouvez utiliser une application telle que PuTTY, mais ce n'est pas obligatoire. Si vous utilisez Windows 10 ou Windows 11, il y a un client SSH natif pour vous permettre de vous connecter à votre NAS.

Ouvrez une invite de commande et saisissez la commande "**ssh**" selon le modèle suivant :

```
ssh <nom d'utilisateur>@<adresse ip>
```

Par exemple, pour se connecter sur le NAS avec l'adresse IP "**192.168.1.200**" avec le compte "**itconnect**" :

```
ssh itconnect@192.168.1.200
```

Si vous avez besoin de préciser un numéro de port spécifique (autre que le port 22), ajoutez cette option en ajustant le numéro de port :

```
ssh itconnect@192.168.1.200 -p 222
```

Saisissez votre mot de passe et vous devriez **avoir accès au shell du système DSM.** Attention, le compte que vous utilisez pour vous connecter doit être **membre du groupe "administrators" de DSM** (c'est le cas du compte créé nativement lors de l'installation du NAS).

Une fois que vous avez accès à votre NAS en ligne de commande, vous devez utiliser la commande "**id**".

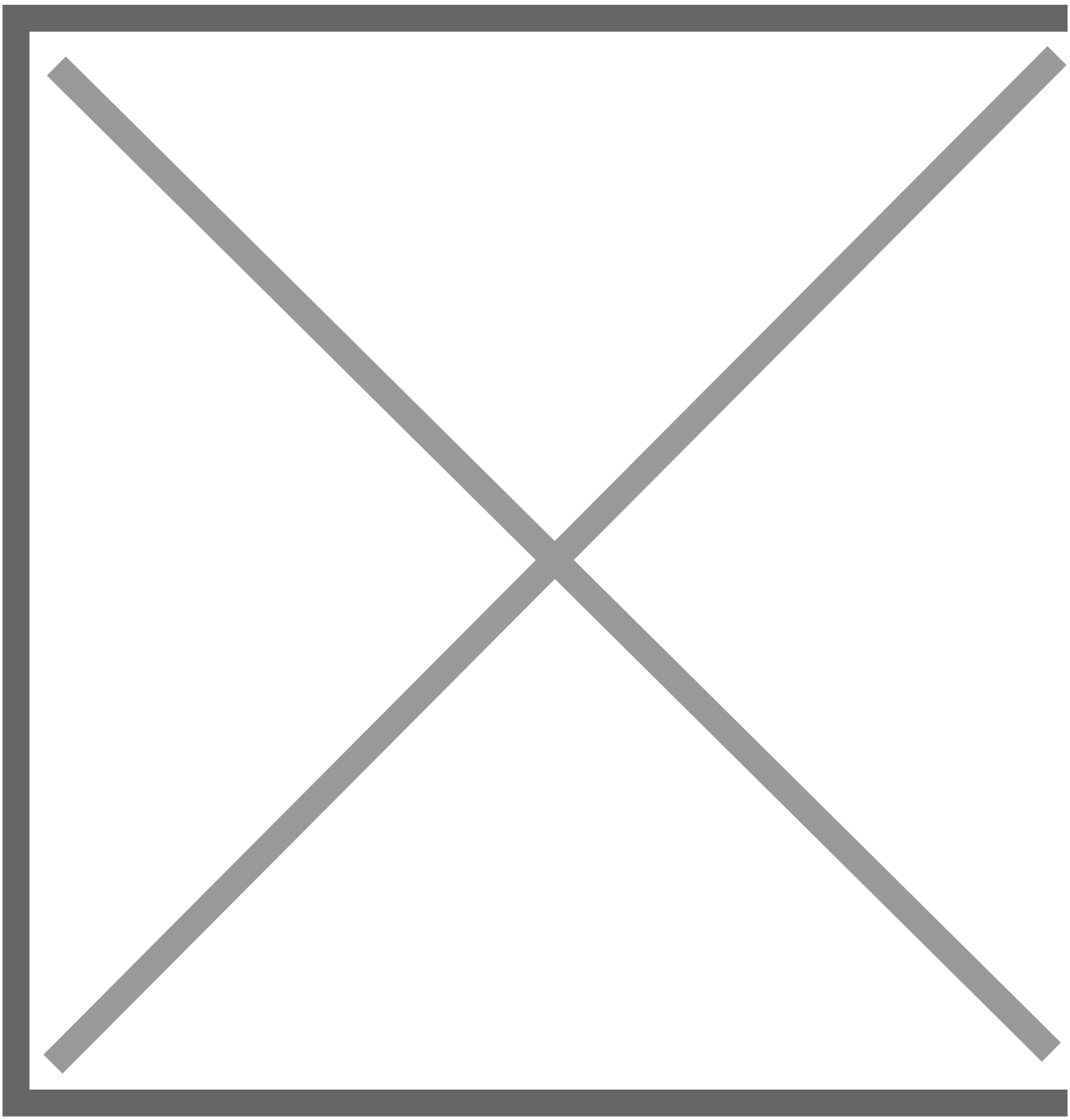
- **Pour récupérer l'UID et le GID du compte avec lequel vous êtes connecté en SSH :**

```
id
```

- **Pour récupérer l'UID et le GID d'un autre compte existant sur votre NAS (exemple avec le compte nommé "docker")**

```
id docker
```

Ici, nous pouvons voir que l'UID de cet utilisateur est "**1027**" tandis que son GID est "**100**". Ici, c'est bien le **groupe principal auquel appartient l'utilisateur** qui est retourné, car il peut être membre de plusieurs groupes.



Il ne reste plus qu'à faire bon usage de ces deux informations !

Conclusion

Grâce à cette astuce, vous êtes en mesure de récupérer l'UID et le GID d'un compte utilisateur de votre NAS Synology ! Ces informations étant différentes d'un NAS à un autre et d'un utilisateur à un autre, c'est une manipulation à connaître.

Comment réinitialiser mon Synology NAS

Cet article fournit des détails et des instructions pour deux options de réinitialisation. Si vous avez oublié votre mot de passe, souhaitez déplacer votre Synology NAS vers un autre environnement réseau ou devez attribuer une nouvelle adresse IP, utilisez le **Mode 1**. Si vous souhaitez réinitialiser votre appareil en réinstallant DSM, utilisez le **Mode 2**.

Environnement

Assurez-vous que le statut de votre Synology NAS est **Prêt** via l'un des outils suivants :

1. **Web Assistant** : Entrez **find.synology.com** dans la barre de recherche de votre navigateur web et trouvez votre Synology NAS.
2. **Synology Assistant** : Ouvrez l'utilitaire de bureau et trouvez votre Synology NAS, disponible dans le Centre de téléchargement.

Solution

Mode 1 : Réinitialiser les identifiants de connexion administrateur et les paramètres réseau

1. Localisez le bouton **RESET** sur votre Synology NAS.
2. Utilisez un trombone pour appuyer doucement et maintenir enfoncé le bouton **RESET** pendant environ 4 secondes jusqu'à ce que vous entendiez un bip, puis relâchez immédiatement le bouton.
3. Lancez **Web Assistant**. Double-cliquez sur votre Synology NAS. Sur la page de connexion, entrez le nom d'utilisateur par défaut du système **admin**, laissez le champ du mot de passe vide, et cliquez sur **Connexion**.
4. Créez un mot de passe fort et cliquez sur **Soumettre**.
5. Connectez-vous à DSM avec le nom d'utilisateur **admin** et le mot de passe que vous venez de créer.
6. Accédez à **Panneau de configuration** > **Utilisateur** > l'onglet **Utilisateur** et double-cliquez sur le compte administrateur que vous souhaitez utiliser.
7. Dans la fenêtre contextuelle, allez à l'onglet **Info** et réinitialisez votre mot de passe. Cliquez sur **OK**.

8. Désactivez le compte admin en vous connectant d'abord avec votre compte administrateur, puis en allant dans **Panneau de configuration** > **Utilisateur** > l'onglet **Utilisateur**. Double-cliquez sur **admin** et cochez la case **Désactiver ce compte**. Cliquez sur **OK**.
9. Le statut du compte **admin** devrait maintenant être **Désactivé**.

Mode 2 : Réinitialiser le Synology NAS et réinstaller le système d'exploitation DSM

Ce mode efface toutes les configurations système et effectue toutes les fonctions de réinitialisation incluses dans le Mode 1.

1. Trouvez le bouton **RESET** à l'arrière de votre Synology NAS. Si vous avez des difficultés à localiser le bouton RESET, sélectionnez votre modèle de Synology NAS sur cette page, allez à l'onglet **Documents** et consultez le **Manuel du produit**.
2. Utilisez un trombone pour appuyer doucement et maintenir enfoncé le bouton **RESET** pendant environ 4 secondes jusqu'à ce que vous entendiez un bip, puis relâchez immédiatement le bouton.
3. Dans les 10 secondes, appuyez à nouveau sur le bouton **RESET** et maintenez-le enfoncé pendant 4 secondes jusqu'à ce que vous entendiez 3 bips supplémentaires.
4. Attendez environ 2 minutes jusqu'à ce que le voyant STATUS de votre Synology NAS clignote en orange, indiquant que votre Synology NAS a été réinitialisé avec succès et que les configurations système ont été effacées.
5. Pour réinstaller DSM, entrez **find.synology.com** dans la barre de recherche de votre navigateur web et trouvez votre Synology NAS sur la page **Web Assistant**. Le nom de serveur de votre Synology NAS devrait être soit DiskStation, FlashStation, ou RackStation, selon votre modèle de Synology NAS. Le statut de votre appareil devrait être **Configuration perdue**.
6. Double-cliquez sur votre Synology NAS et suivez l'assistant pour terminer le processus de réinstallation.

> Remarques :
> 1. Le processus de réinitialisation sur un Synology NAS ne fonctionnera que si un disque est installé et que DSM est configuré.
> 2. Si vous souhaitez uniquement changer le mot de passe de l'administrateur, allez dans **DSM** > **Options** > **Personnel** > **Compte** pour le faire sans réinitialiser votre Synology NAS.
> 3. La réinitialisation du Synology NAS n'affecte pas ses données. Cependant, nous recommandons fortement de lancer **Backup & Replication** (disponible sur DSM 5.2) ou **Hyper Backup** (disponible à partir de DSM 6.0) pour sauvegarder les données et les configurations système avant la réinitialisation.
> 4. Selon cet avis de fin de disponibilité, Synology n'offre plus DSM 6.2.3 et les versions

antérieures sur son site Web. Avant de réinitialiser, assurez-vous d'avoir le fichier d'installation nécessaire, car il ne sera pas téléchargé automatiquement pendant le processus.

5. Consultez la page d'aide de Synology Assistant pour les définitions des statuts des appareils.
6. Une fois votre Synology NAS réinitialisé via le Mode 1, les paramètres suivants s'appliquent :
- Le compte **admin** est restauré par défaut.
- Le port de gestion de l'interface utilisateur est réinitialisé à 5000/5001.
- Les interfaces IP, DNS, passerelle et autres interfaces réseau sont réinitialisées à DHCP.
- PPPoE est désactivé.
- Le blocage automatique est désactivé.
- Les règles de pare-feu sont désactivées.
- Le cluster haute disponibilité est supprimé.
- Le cluster Virtual Machine Manager est supprimé.
- Les dossiers chiffrés sont démontés et la fonction **Monter automatiquement au démarrage** est désactivée. Pour des raisons de sécurité, si le chiffre est une clé machine, elle est supprimée du Gestionnaire de clés. Apprenez comment conserver la clé machine pour déchiffrer les dossiers partagés après la réinitialisation.

7. Consultez le manuel du produit si vous avez besoin d'aide pour localiser le bouton RESET.
8. Si vous avez précédemment défini **admin** comme nom d'utilisateur pour votre compte administrateur, vous pouvez simplement utiliser ce nom d'utilisateur avec le mot de passe fort que vous venez de définir pour vous connecter à DSM.

Container Manager

Homer : Un tableau de bord personnalisable

Dans ce tutoriel, nous allons apprendre à déployer l'application Homer sur un NAS Synology, à l'aide d'un conteneur Docker ! Homer est un projet open source destiné à l'auto-hébergement dont l'objectif est de vous permettre de déployer **une page d'accueil aux allures de tableau de bord sur votre propre serveur** !

Sur cette page statique, vous allez pouvoir **ajouter tous les éléments et liens** que vous jugez nécessaire ! Par exemple, vous pouvez **lister vos sites favoris, ajouter des liens vers vos applications préférées, ou encore vers vos équipements** ! Il y a aussi la possibilité de remonter des informations à partir de services personnalisés, c'est-à-dire d'autres applications (Prometheus, AdGuard Home, Portainer, PiHole, Proxmox, etc...).

Homer peut s'avérer utile dans de nombreux scénarios et cette page est facilement personnalisable grâce à un fichier de configuration au format YAML.

image.png

Installer l'application Homer sur son NAS

Avant de créer le conteneur, nous allons préparer un répertoire pour stocker ses données. Au sein du répertoire "**docker**", nous allons créer le répertoire "**homer**" afin de maintenir la logique habituelle : **un répertoire par conteneur**. Ce qui donne :

image.png

Puis, dans le répertoire "**homer**", nous allons créer un répertoire "**data**" qui sera utilisé pour stocker les données applicatives d'Homer. Ce qui donne :

image.png

Désormais, nous pouvons lancer l'application Container Manager (Docker) pour **créer un nouveau conteneur à partir d'un code de configuration Docker Compose**.

Dans "**Container Manager**", cliquez sur "**Projet**" puis sur "**Créer**". Nommez ce projet "**homer**" puis indiquez le répertoire "**/docker/homer**" comme chemin pour ce conteneur. Autrement dit, l'option "**Chemin**" doit avoir pour valeur "**/docker/homer**".

image.png

En ce qui concerne la "**Source**", choisissez l'option "**Créer un fichier docker-compose.yml**". Une zone de texte apparaît : qu'allons-nous écrire ici ? Nous allons récupérer le code du fichier "docker-compose.yml" disponible sur GitHub officiel pour ensuite l'adapter.

Ce qui donne :

```
---
version: "2"
services:
  homer:
    image: b4bz/homer
    container_name: homer
    volumes:
      - /volume1/docker/homer/data:/www/assets
    ports:
      - 8080:8080
    user: 1000:1000 # default
    environment:
      - INIT_ASSETS=1 # default
```

Désormais, nous allons devoir modifier deux options : le chemin vers le répertoire local pour mapper le répertoire "**/www/assets**" du conteneur vers "**/volume1/docker/homer/data**", et les informations sur le compte utilisateur à utiliser pour exécuter le conteneur. Ici, l'utilisateur "**docker** " de mon NAS est spécifié en indiquant son UID "**1027**" et son GID "**100**".

Ici, nous ne modifions pas le mappage sur le numéro de port, donc **l'application sera accessible sur le port 8080**. Vous pouvez l'adapter si besoin. De plus, l'option "**INIT_ASSETS=1**" permet d'ajouter les fichiers de démonstrations à l'application, ce qui évite de partir de zéro.

> Pour le nom de l'image, vous pouvez ajouter le tag "**latest**" pour récupérer la dernière image Docker de ce projet associée à ce tag. Ce qui donne la valeur "**b4bz/homer:latest**" pour la directive "**image**".

image.png

Une fois le fichier Docker Compose prêt, vous pouvez continuer jusqu'à la fin pour créer le conteneur. L'image "**b4bz/homer**" sera téléchargée à partir du Docker Hub et utilisée pour exécuter le conteneur.

Voilà, le conteneur Docker "homer" est actif !

image.png

Dès à présent, nous pouvons accéder à l'interface de l'application :

```
http://<adresse IP du NAS>:8080
```

Vous devriez obtenir ceci :

image.png

Désormais, nous allons évoquer la personnalisation de cette page d'accueil.

Personnaliser la page d'accueil Homer

Pour modifier la page d'accueil d'Homer, nous allons éditer le fichier suivant :

```
/docker/homer/data/config.yml
```

Pour l'éditer directement depuis l'interface de DSM, installez l'application "**Éditeur de texte**" depuis le "**Centre de paquets**". Ceci permet d'ajouter l'option "**Ouvrir avec un éditeur de texte**" dans le menu contextuel de DSM afin d'éditer les fichiers en ligne.

image.png

Le fichier de configuration s'ouvre. **Il s'agit d'un fichier au format YAML**, donc il faut respecter rigoureusement l'indentation, les espaces, etc... Pour ne pas générer de problèmes de syntaxes. L'édition est assez simple puisque le code est facilement lisible.

image.png

Vous pouvez charger vos images dans le répertoire "**data/tools**" de votre conteneur pour les appeler en tant que logo dans Homer. Par ailleurs, **Homer s'appuie sur la bibliothèque FontAwesome pour charger les icônes**, donc utilisez cette page pour rechercher un logo à intégrer sur une entrée (par exemple "**fas fa-hdd**" pour l'icône en forme de disque dur).

À titre d'exemple, voici le code du nœud « services » qui permet d'obtenir le résultat présenté ci-dessus avec les deux blocs "**Mes sites favoris**" et "**Mes NAS**".

```
# Services
# First level array represent a group.
# Leave only a "items" key if not using group (group name, icon & tagstyle are optional, section separation will not be displayed).
services:
  - name: "Mes sites favoris"
    icon: "fas fa-cloud"
```

```
    items:
      - name: "IT-Connect"
        logo: "assets/tools/Logo-IT-Connect.png"
        subtitle: "Tutoriels, cours, actualités - Informatique"
        tag: "tutos"
        keywords: "tutos"
        url: "https://www.it-connect.fr"
        target: "_blank" # optional html a tag target attribute
  - name: "Mes NAS"
    icon: "fas fa-hdd"
    items:
      - name: "NAS Synology DS220+"
        icon: "fas fa-hdd"
        subtitle: "https://192.168.1.200:5001"
        tag: "nas"
        keywords: "nas"
        url: "https://192.168.1.200:5001"
        target: "_blank" # optional html a tag target attribute
      - name: "NAS Synology DS923+"
        icon: "fas fa-hdd"
        subtitle: "https://192.168.1.201:5001"
        tag: "nas"
        keywords: "nas"
        url: "https://192.168.1.201:5001"
        target: "_blank" # optional html a tag target attribute
```

Quand vous effectuez une modification, enregistrez le fichier et rafraichissez la page web pour voir ce que ça donne. Homer étant une page d'accueil statique, elle est très rapide à charger et très légère.

NAS Synology : comment sauvegarder et restaurer un container Docker ?

Dans ce tutoriel, nous allons apprendre à sauvegarder et restaurer un container Docker qui tourne un NAS Synology ! Sur les NAS, il est de plus en plus fréquent de mettre en place des applications et services à partir d'un container Docker : mais qu'en est-il de la sauvegarde ? Si le container plante totalement, comment le remettre en service sans perdre de données ? Et comment procéder à la restauration ? Voici mes conseils sur le sujet !

Lorsque l'on crée un container Docker, au-delà de l'image du container en lui-même, il convient de définir **un emplacement de stockage** pour que le container soit en mesure de **stocker ses données** : c'est **ce répertoire qu'il faut sauvegarder** ! Il n'existe pas encore d'outil pour sauvegarder l'intégralité du container avec ses données : on va devoir procéder en deux temps. La sauvegarde des données d'une part, et la sauvegarde de la configuration du container d'autre part.

Lorsque l'on va restaurer un conteneur Docker, on ne va pas réellement restaurer une sauvegarde ! En effet, on va créer un nouveau container avec la même configuration que le container HS, et restaurer les données au même emplacement (à partir de la sauvegarde de données).

Pour cet exemple, c'est le container Docker "Vaultwarden" qui va servir de cobaye !

> **Remarque** : depuis DSM 7.2, l'application Docker proposée par Synology s'appelle Container Manager. Toutefois, la méthode évoquée dans cet article fonctionne dans les deux cas.

Sauvegarder les containers Docker

Sauvegarde des données avec Hyper Backup

Commençons par évoquer la sauvegarde des données. À partir de la console Container Manager, dans les détails du conteneur "vaultwarden", on peut avoir des informations sur le stockage dans la section "**Volume**". Ici, on voit que le conteneur stocke ses données à l'emplacement suivant sur le NAS : "**/volume1/docker/vaultwarden**. C'est ce répertoire qu'il va falloir sauvegarder, enfin, plutôt le dossier racine "Docker" pour sauvegarder les données de tous les conteneurs (à condition d'utiliser cette racine pour tous vos conteneurs).

Synology - Volume d'un container Docker

À partir de l'explorateur de fichiers File Station de DSM, on peut voir que ce répertoire contient les données du conteneur "vaultwarden" :

DSM File Station - Docker exemple données

Il ne reste plus qu'à créer une tâche de sauvegarde avec l'**application Hyper Backup** développée par Synology. On doit l'installer à partir du **Centre de paquets**.

Ensuite, on doit ouvrir l'application une fois qu'elle est installée et créer une nouvelle tâche en cliquant sur le "+".

Installer Hyper Backup sur DSM 7.2

À la première étape "**Sélectionner un type de sauvegarde**", on sélectionne "**Dossiers et paquets**".

Hyper Backup est capable de réaliser les sauvegardes en local, vers un périphérique de stockage connecté en USB, vers un autre NAS, ou encore dans le Cloud. En fonction de votre configuration, choisissez l'option qui vous convient le mieux. Dans mon cas, et pour cet exemple, la destination sera un dossier local alors je sélectionne "**Dossier local & USB**". Idéalement, il faudrait prévoir aussi une sauvegarde externalisée.

Hyper Backup - Sauvegarde container Docker en local

À l'étape suivante, on doit sélectionner l'emplacement des sauvegardes. Ici, ce sera au sein du dossier partagé nommé "**Sauvegardes**" au sein d'un répertoire nommé "**Docker**" (qui ne doit pas exister).

Hyper Backup - Tâche de sauvegarde Docker

À l'étape suivante, il convient de sélectionner les données à sauvegarder : **on sélectionne le répertoire "docker" car il contient un sous-répertoire par container** (selon la configuration adoptée sur mon NAS). On pourrait sélectionner uniquement le répertoire d'un container spécifique.

Sélectionner les données à sauvegarder dans Hyper Backup

Passez l'étape "**Sauvegarde des applications**" car il n'est pas possible de choisir Container Manager dans la liste.

L'étape "**Paramètres de sauvegarde**" permet de **nommer la tâche** et d'indiquer **à quelle fréquence** on souhaite effectuer la sauvegarde des données du container.

Paramètres de la sauvegarde Docker Synology

Poursuivez. À l'étape "**Paramètres de rotation**", on doit indiquer le nombre de sauvegardes à conserver. Il est conseillé d'**activer la rotation pour gérer l'espace de stockage**, sinon l'espace de stockage consommé par les sauvegardes ne fera qu'augmenter. S'il s'agit d'une sauvegarde Cloud, la facture pourrait être salée... Dans l'exemple ci-dessous, on conserve 30 versions et comme il y aura une sauvegarde par jour, ceci me permet d'avoir **un historique sur 30 jours**.

Rotation des sauvegardes Docker Synology

Validez la création de la tâche... On est invité à lancer la tâche dès maintenant : bonne idée pour tester son bon fonctionnement ! **La sauvegarde des données des container est un succès !**

Sauvegarde Docker réussie dans Hyper Backup

Sauvegarde de la configuration du container

En complément de la sauvegarde des données du container, il est important de sauvegarder sa configuration. **Cette opération ne peut pas être automatisée depuis l'interface de DSM.** La bonne nouvelle, c'est qu'une fois qu'un container est en place, on modifie rarement sa configuration... Le fait de sauvegarder la configuration va permettre d'obtenir **un fichier JSON** qui contient la déclaration du volume, des ressources, des ports utilisés, etc... Ainsi, **on peut recréer le container en l'état en quelques clics !** C'est l'objectif.

À partir de Container Manager, on sélectionne le container "**vaultwarden**", on clique sur "**Action**" puis sur "**Exporter**".

Sauvegarder configuration Container Docker Synology

Ici, on a la possibilité d'**exporter les paramètres du conteneur** vers l'**ordinateur local** (ou vers un répertoire du NAS).

Exporter un container Docker Synology

Ceci va permettre de sauvegarder la configuration du container !

Restaurer un container Docker

Imaginons que le container "vaultwarden" soit inutilisable : **que faire pour le remettre en service ? Comment le restaurer ?** Pour simuler une panne, j'ai supprimé le container de mon NAS et j'ai renommé le répertoire "vaultwarden" (on peut aussi le supprimer directement).

Suppression d'un container Docker DSM

Créer un nouveau container propre

Tout d'abord, il convient de créer un conteneur propre, mais l'on ne partira pas de zéro ! On va **importer le fichier JSON qui contient la configuration du conteneur** ! À partir de Container Manager, on clique sur "**Conteneur**", puis sur "**Action**", "**Importer**" puis "**À partir du périphérique local**".

Importer la configuration d'un container Docker Synology

On sélectionne le fichier "vaultwarden.json", on nomme le conteneur et on clique sur "**Importer**".

Importer le conteneur Docker DSM

Le conteneur Docker est créé, avec la même configuration qui existait précédemment ! De ce fait, il pointe déjà vers le répertoire "**/volume1/docker/vaultwarden**" du NAS, mais il lui manque ses données ! Avant de le démarrer, on va restaurer ses données.

Restaurer les données du container

La restauration des données s'effectue à partir de l'interface Hyper Backup. On sélectionne la tâche de sauvegarde "**Sauvegarde locale Docker**" sur la gauche (1), puis on clique sur "**Liste des versions**" (2) avant de cliquer sur le **bouton Backup Explorer** (3) après avoir sélectionné la version à restaurer.

Hyper Backup - Liste des versions de la sauvegarde Docker

On explore la sauvegarde... À la recherche du répertoire "**vaultwarden**" que l'on va simplement sélectionner avant de cliquer sur "**Restaurer**". On valide l'opération pour **restaurer le dossier à son emplacement initial**. Le bouton "Copier vers..." permet de faire la restauration à un autre endroit.

Restaurer les données du conteneur Docker

Une fois le répertoire restauré, il ne reste plus qu'à sélectionner le conteneur et à le démarrer via le menu "**Action**".

Démarrer un container Docker Synology après restauration des données

Voilà, le conteneur Vaultwarden est actif, tout comme l'application qui est de nouveau opérationnelle ! Les données sont bien celles présentes dans l'application au moment de la sauvegarde Hyper Backup.

Conclusion

Si vous utilisez des containers Docker sur votre NAS Synology, je vous encourage vivement à mettre en place la sauvegarde des données de vos conteneurs ! N'attendez

pas un crash avant de vous poser la question : désormais vous avez le tutoriel pour le faire à l'aide d'Hyper Backup. Vous pouvez utiliser une autre application pour la sauvegarde des données si vous préférez.

FreshRSS : Agrégateur de flux RSS

Dans ce tutoriel, nous allons installer la solution FreshRSS sur un NAS Synology afin d'avoir un gestionnaire de flux RSS hébergé en local sur notre NAS ! L'installation sur une machine sous Linux (Debian, Ubuntu, etc.) est possible également.

Pour cette démonstration, je vais utiliser un NAS Synology avec DSM 7.1, en utilisant le paquet Docker pour mettre en place FreshRSS dans un container.

Mais au fait, c'est quoi FreshRSS ? Il s'agit d'un gestionnaire de flux RSS libre et open source, que vous pouvez héberger vous-même sur un NAS, un serveur, un ordinateur, etc... Sur le site officiel, on peut lire "*FreshRSS est un agrégateur et lecteur de flux RSS. Il permet de regrouper l'actualité de plusieurs sites différents dans un endroit unique pour que vous puissiez la lire sans devoir aller de site en site.*"

Bien qu'il y ait d'autres alternatives diverses et variées (QuiteRSS, Netvibes, Inoreader, Feedly, voire même Outlook peut accueillir des flux RSS), FreshRSS est à mon sens l'un des meilleurs agrégateurs de flux RSS disponibles et que l'on peut auto-héberger.

Un gestionnaire de flux RSS est très intéressant pour faire une bonne veille technologique. En effet, dans une même interface, vous allez pouvoir **visualiser les derniers articles mis en ligne par vos sites préférés** ! Quand un nouvel article est ajouté, le flux RSS est mis à jour et donc l'information remonte dans votre interface FreshRSS. Ainsi, vous n'avez pas besoin d'aller sur chacun des sites : si un article vous intéresse, vous cliquez dessus, et vous serez redirigé vers le site en question, sur la page de l'article.

Au-delà de l'utiliser pour **suivre les articles publiés sur divers sites** (organisés par catégorie), vous pouvez l'utiliser pour **suivre la sortie du release sur un projet GitHub** (par exemple, une nouvelle version de PowerShell), ou encore **une nouvelle alerte de sécurité émise par le CERT-FR** ou un autre organisme. Le tout, c'est que le site / service mette à disposition des utilisateurs un flux RSS.

Mise en place du container FreshRSS

Commençons par **créer un dossier nommé "freshrss"** (attention à la casse) **dans le répertoire "docker"** de votre NAS à l'aide de l'explorateur File Station. Ce répertoire sera utilisé par le conteneur pour stocker la configuration de l'application. Vous pouvez aussi le créer en même

temps que le conteneur.

Dans ce dossier, créez un répertoire nommé "**extensions**" : on l'utilisera pour stocker les extensions de FreshRSS.

image.png

Ensuite, il faut basculer sur Docker pour créer le conteneur. Cliquez sur "**Conteneur**" à gauche `1`, puis sur le bouton "**Créer**" `2`. Cliquez sur "**Ajouter**" `3` puis "Ajouter à partir d'une URL".

image.png

Ici, vous indiquez l'URL "**freshrss/freshrss**" ce qui va permettre de rechercher l'image sur Docker Hub directement. Cliquez sur "**Ajouter**" `1`, la fenêtre "**Choisir un identification**" va apparaître donc prenez la dernière version de FreshRSS via "**latest**" `2`. Validez via le bouton "**Sélectionnez**" `3`.

Remarque : preuve que cette image est officielle, on peut aussi effectuer le téléchargement à partir de la section "Registre" dans l'interface Docker.

image.png

Pour le réseau, conserver en mode "**Bridge**" car nous avons besoin que le conteneur utilise le réseau du NAS et nous devons modifier les associations de port (*port interne au conteneur VS port externe au niveau du NAS*).

image.png

En ce qui concerne les paramètres généraux... Commencez par donner un nom au conteneur, par exemple "*freshrss*" ou "*syno-freshrss*". Puis, cochez l'option "**Activer le redémarrage automatique**". Il n'est pas nécessaire de toucher aux autres options.

image.png

Avant de continuer, cliquez sur "**Paramètres avancés**" dans le but de définir quelques variables d'environnement propre à notre conteneur. Il y a 4 variables à définir :

- **TZ** : le fuseau horaire, donc ici "Europe/Paris" pour la France Métropolitaine
- **CRON_MIN** : quand faut-il rafraichir les flux RSS ? Ici, deux fois par heure, à la minute "10" et à la minute "40"
- **FRESHRSS_ENV** : mode "production"
- **LISTEN** : port local sur lequel écouter, on reste sur la valeur par défaut à savoir 80

image.png

Cliquez sur "**Sauvegarder**".

Passons à l'étape "**Paramètres des ports**". Pour le port du conteneur, vous devez conserver "**80**" car cela correspond au protocole HTTP. Par contre, pour le "**Port local**" vous devez indiquer **un numéro de port qui n'est pas encore utilisé sur votre NAS**. Par exemple, n'utilisez pas le port "8000" si vous l'utilisez pour un autre conteneur ou une autre application. Ici, je pars sur le numéro de port 8080. Cela signifie que FreshRSS sera accessible à cette adresse :

```
http://<IP du NAS>:8080
```

image.png

Comme je le disais précédemment, FreshRSS a besoin d'un dossier dans lequel stocker ses données, et surtout il est intéressant pour nous de pouvoir y accéder (pour déployer une extension FreshRSS, par exemple). C'est pour cette raison que nous avons créé le répertoire "freshrss" dans "docker".

Une fois à l'étape "**Paramètres du volume**", cliquez sur "**Ajouter un dossier**", sélectionnez le répertoire créé précédemment. Appliquez la configuration suivante :

- Associez "**/docker/freshrss**" à "**/var/www/FreshRSS/data**" pour les données globales de FreshRSS
- Associez "**/docker/freshrss/extensions**" à "**/var/www/FreshRSS/extensions**" pour stocker les extensions

image.png

Voilà, nous sommes à la dernière étape de la création du conteneur. Cliquez sur "**Effectué**" et le tour est joué !

Cette action a pour objectif de déployer le container Docker "FreshRSS" sur le NAS !

Si vous avez configuré le pare-feu de DSM sur votre boîtier, vous devez **créer une règle pour autoriser les accès sur le port "8080"**. Pour rappel, voici le chemin pour accéder à la configuration du pare-feu : **Panneau de configuration > Sécurité > Pare-feu**.

image.png

Ensuite, vous pouvez également visualiser le container dans l'interface de Docker puis en cliquant sur "**Conteneur**". Vous devriez voir ceci :

image.png

Le container Docker FreshRSS est prêt : il héberge un serveur Web et les sources d'installation de l'application. Désormais, nous devons finaliser l'installation de FreshRSS.

Installation de FreshRSS

À partir d'un navigateur, **accédez à votre NAS sur le port "8080"** en précisant l'adresse IP de votre boîtier. Une devez arriver sur l'étape 1 de l'installation de FreshRSS. Commencez par choisir la langue et validez.

```
http://<IP du NAS>:8080
```

La première étape s'affiche !

image.png

Ensuite, il y a **la vérification des dépendances.** Ici, tous les feux sont au vert normalement ! Donc, poursuivez.

image.png

L'étape 3 consiste à choisir le moteur de base de données. **Le type de base de données SQLite me semble suffisant pour un agrégateur de flux RSS à usage personnel.** Il n'y a pas besoin de performances folles, même si quelques utilisateurs se connectent dessus. Sinon, il y a toujours possibilité de se connecter sur un serveur MySQL ou PostgreSQL.

image.png

A l'étape 4, définissez un nom d'utilisateur et un mot de passe. Ce sera votre comte pour vous connecter au gestionnaire de flux RSS.

image.png

Cliquez sur le bouton "**Terminer l'installation**" : FreshRSS est installé !

image.png

Désormais, vous pouvez vous authentifier avec votre compte utilisateur (celui de FreshRSS, pas celui du NAS).

image.png

Voilà, l'interface de FreshRSS s'affiche avec le flux RSS de suivi des versions de FreshRSS inclus par défaut ! **Bienvenue sur votre agrégateur de flux RSS !**

image.png

En cliquant sur la roue crantée en haut à droite, vous pouvez **accéder aux paramètres de l'application**. Par exemple, la partie "**Affichage**" vous permettra de changer le thème. Je vous encourage à faire ce premier changement, car il y a des thèmes beaucoup plus sympas que celui d'origine !

image.png

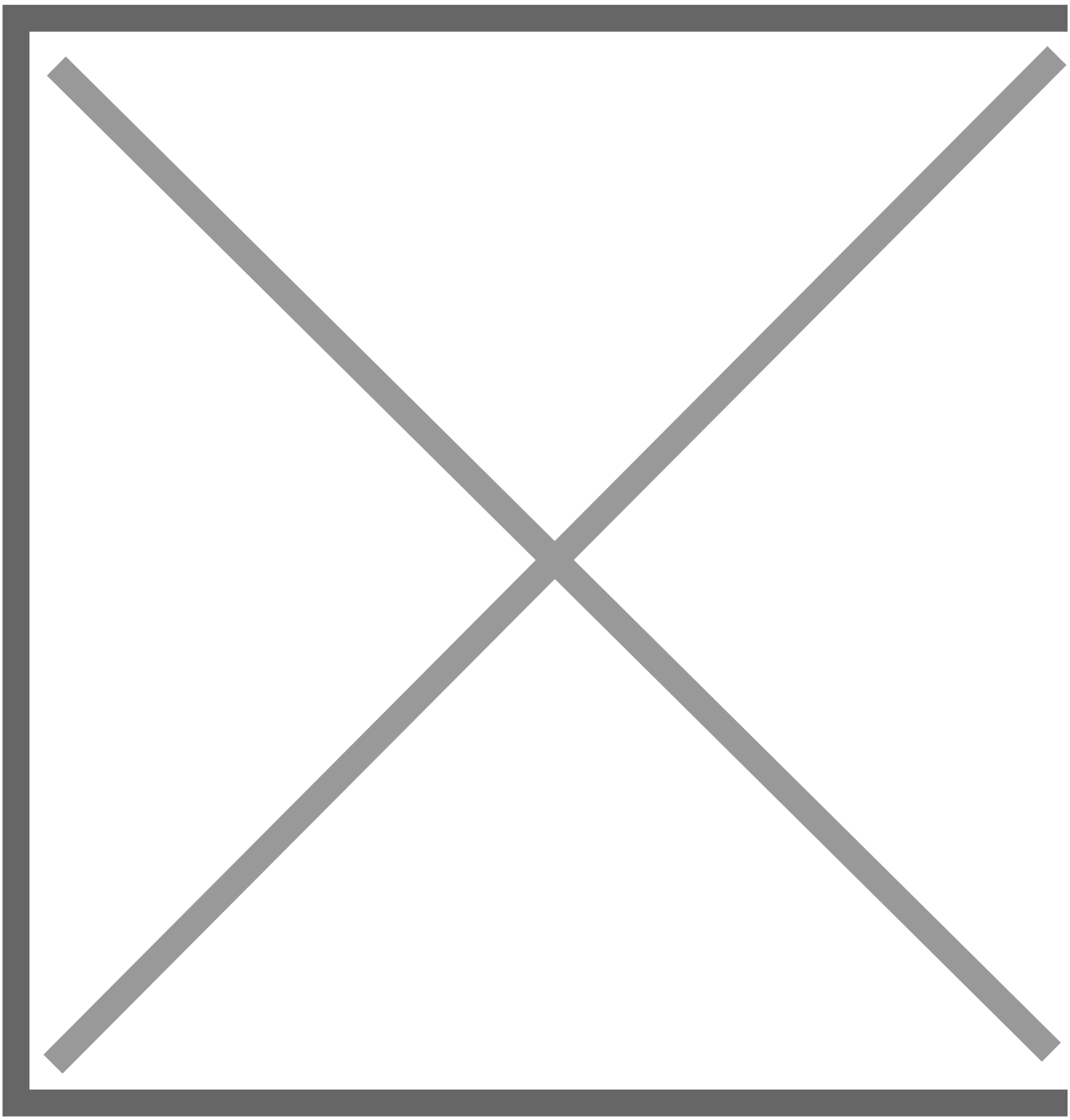
CyberChef : boite à outils cybersécurité

Dans ce tutoriel, nous allons apprendre à installer l'application CyberChef sur un NAS Synology, à l'aide d'un conteneur Docker. Ceci va vous permettre d'héberger une **boite à outils** très utile que vous travaillez dans la **cybersécurité** ou que vous soyez **administrateur système et réseau**. L'avantage de l'héberger sur un NAS Synology, c'est que les **outils CyberChef** pourront être accessibles à **un ensemble d'utilisateurs à partir d'une seule instance** hébergée sur le NAS.

Pour rappel, CyberChef regroupe de nombreux outils pour chiffrer/déchiffrer, encoder/décoder et transformer des données. Il s'agit d'une application mise à disposition par le GCQH (*Government Communications Headquarters*), c'est-à-dire les services de renseignements britanniques.

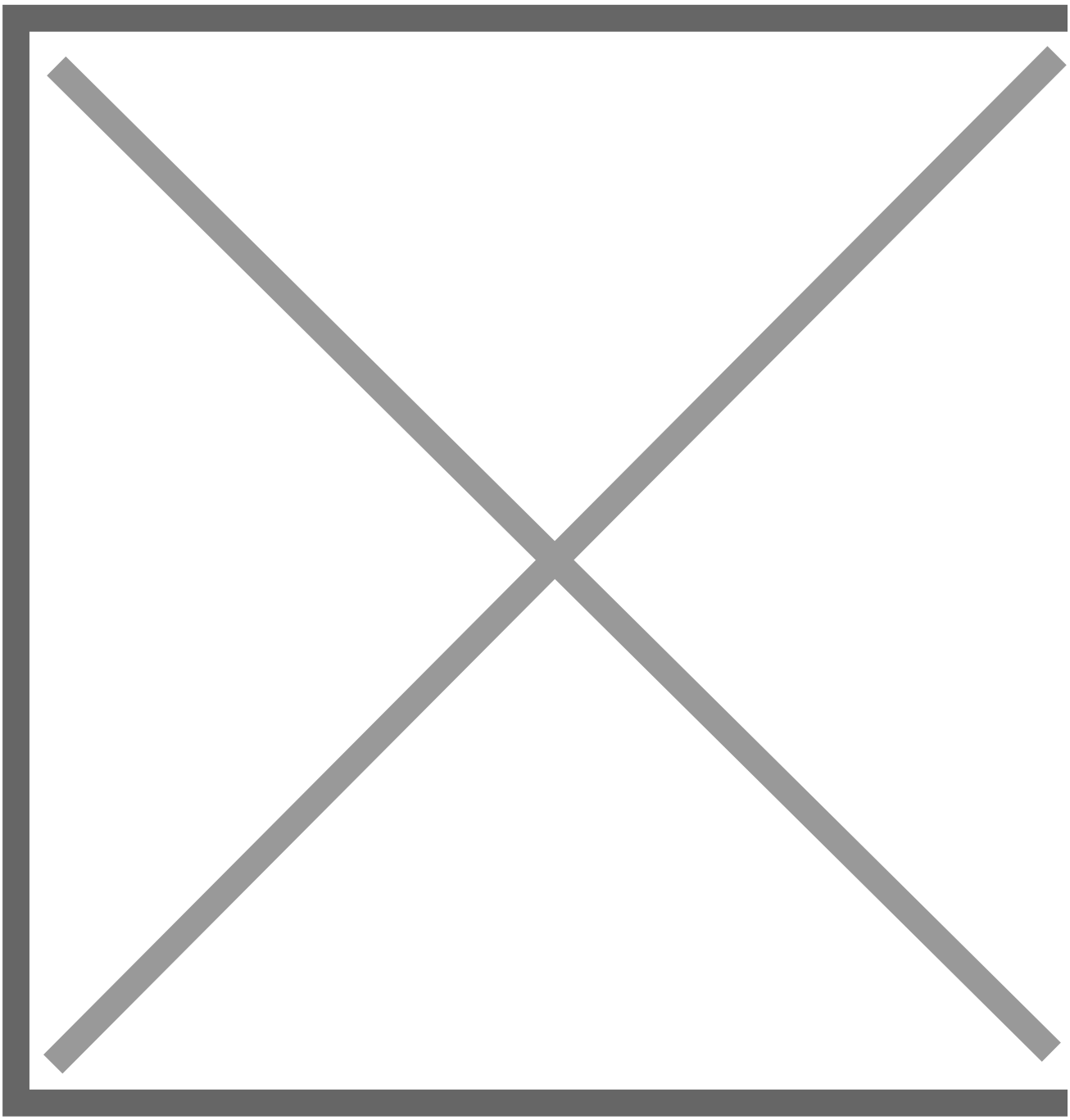
Déployer CyberChef avec Docker

Avant toute chose, vous devez installer l'application "**Container Manager**" sur votre NAS, si ce n'est pas déjà fait. Puis, vous devez créer un dossier pour ce conteneur. Pour ma part, il s'agit du répertoire "**cyberchef**" créé sous "**docker**". Bien que ce répertoire soit nécessaire pour Container Manager, il n'est pas réellement utile pour CyberChef, car cette application ne stocke pas en local les données que vous lui envoyez en traitement. Les traitements sont effectués du côté client, c'est-à-dire dans le navigateur de l'utilisateur.



Ensuite, ouvrez "**Container Manager**" puis cliquez sur "**Projet**" afin de créer un nouveau projet en cliquant sur le bouton nommé "**Créer**".

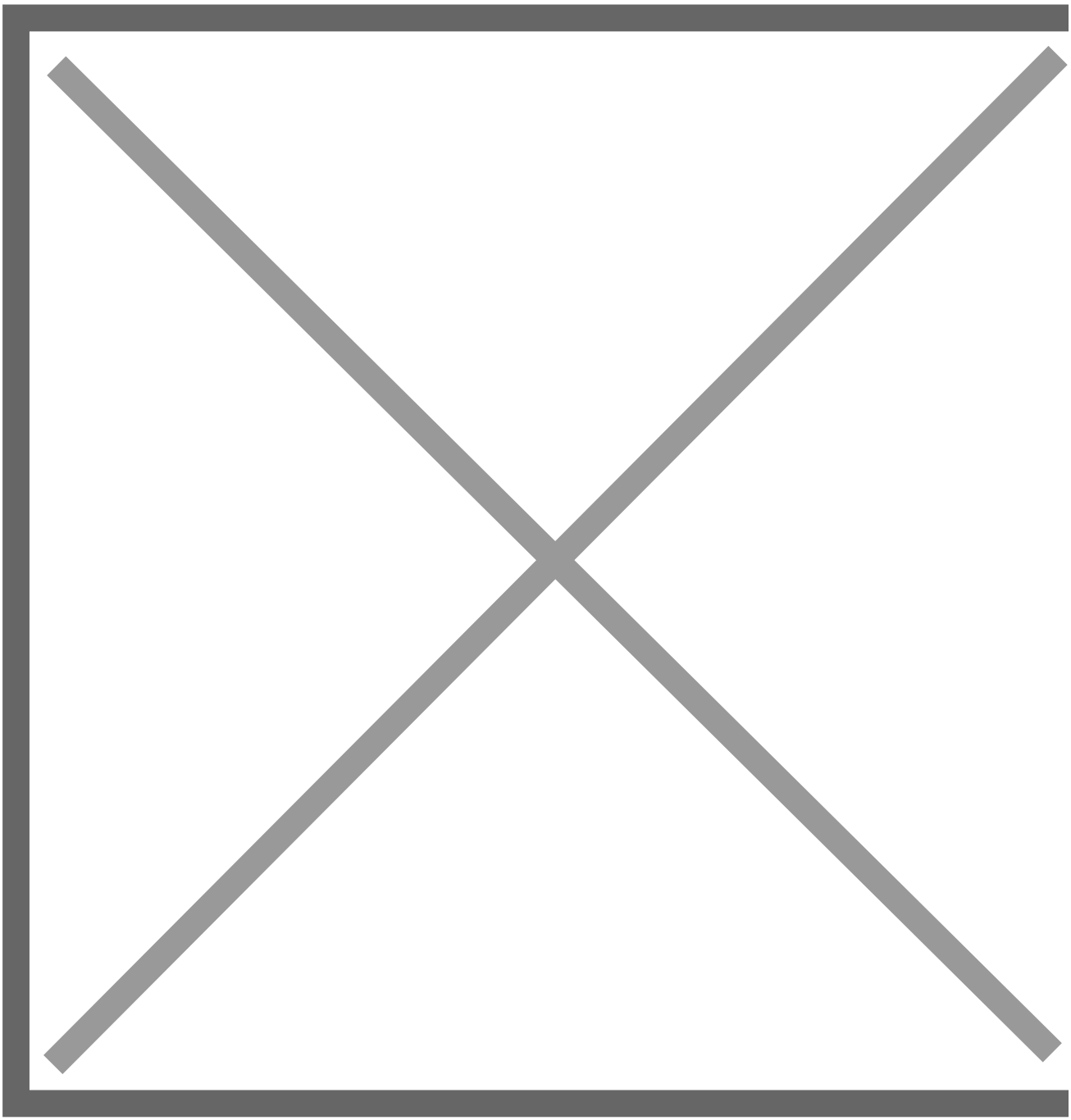
Vous devez donner un nom à ce projet, sélectionner le répertoire créé précédemment et indiquer le code de configuration Docker Compose permettant de déployer l'image officielle du conteneur CyberChef. Ce qui donne :



Voici le code Docker Compose :

```
version: "3"

services:
  cyber-chef:
    image: mpepping/cyberchef:latest
    ports:
      - "8000:8000"
```

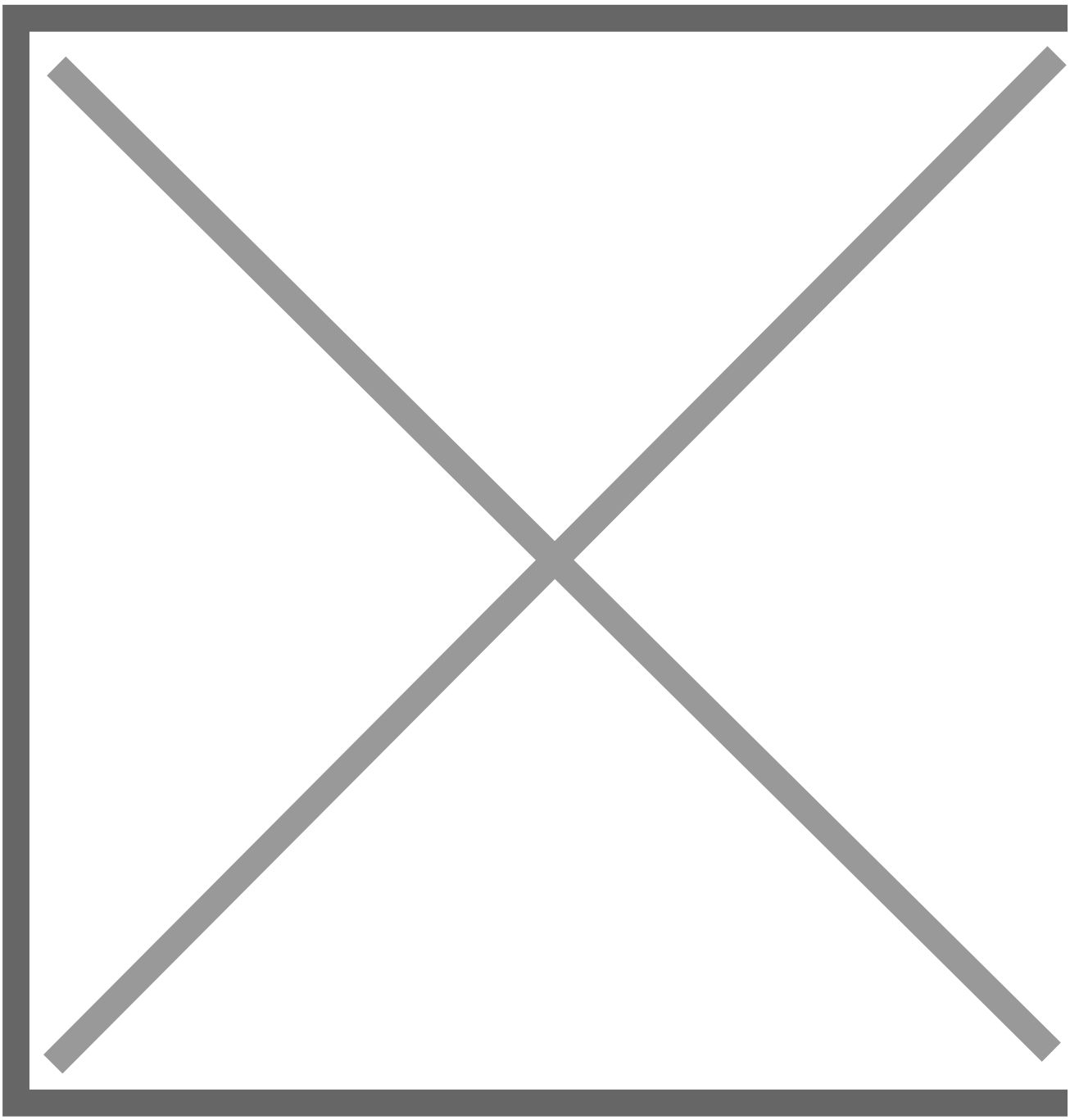
Dans le cas présent, le port externe sur lequel sera joignable l'application, est le **port 8000**. Modifiez la valeur à gauche sur la ligne "**- "8000:8000"**" pour choisir un autre port (facultatif). Puis, poursuivez jusqu'à la fin et lancez la création du projet... Patientez pendant le téléchargement de l'image Docker et la création du conteneur associé.



Dès à présent, vous pouvez accéder à l'application CyberChef de cette façon :

- **http://<adresse IP de votre NAS>:8000**

Vous avez désormais accès à l'application CyberChef, hébergée sur votre NAS ! Il ne vous reste plus qu'à mettre l'URL en favoris et à la transmettre à vos collègues s'il s'agit d'une instance partagée !



Conclusion

Quelques minutes suffisent à déployer l'application CyberChef sur un NAS Synology !
C'est une boite à outils très polyvalente qui peut s'avérer à de nombreux métiers liés à l'informatique. Avec elle, vous n'aurez plus qu'à cuisiner vos données de différentes façons, sans avoir à recourir à la ligne de commande !

Prise en main de Container Manager

Dans ce tutoriel, nous allons effectuer un tour d'horizon de cette application permettant d'exécuter des containers Docker.

Qu'est-ce que Container Manager ?

Le centre des paquets du système d'exploitation DSM contient une application nommée "**Container Manager**". Depuis environ un an et suite à la mise en ligne de DSM 7.2, il s'agit du nouveau de l'application "**Docker**". Son objectif est de vous permettre de **créer et d'exécuter des containers Docker sur votre NAS Synology**, ce qui ouvre de nombreuses possibilités, tout en consommant peu de ressources.

Vous pouvez **créer vos propres containers à partir du référentiel d'images Docker où il y a des milliers d'images différentes**. Ainsi, vous pouvez exécuter sur votre NAS diverses applications, au sein de **containers isolés du système DSM**.

Avant de commencer, vérifiez si votre modèle de NAS Synology est compatible avec le paquet "**Container Manager**" en consultant cette page :

- Synology - Paquet Container Manager

Tour d'horizon de l'interface Container Manager

Installation de Container Manager

La première étape consiste à installer le paquet "**Container Manager**" car il n'est pas préinstallé sur DSM. Vous pourrez le trouver dans le "**Centre de paquets**".

Il suffit de cliquer sur "**Tous les paquets**", de rechercher le terme "**container**" pour localiser l'application "**Container Manager**", puis de cliquer sur le bouton "**Installer**".

image.png

Ensuite, l'application "**Container Manager**" sera accessible dans le menu principal de DSM.

Découverte de l'interface

L'interface de "**Container Manager**" contient un ensemble de sections accessibles dans un menu vertical présent sur la gauche. Lorsque l'application est lancée, nous arrivons dans la "**Vue d'ensemble**", qui est un **tableau de bord** proposant **un aperçu global sur l'état de vos containers** et **les ressources consommées**.

image.png

Pour le reste, voici à quoi correspondent les autres entrées présentes dans le menu latéral :

Projet

La section "**Projet**" est directement liée à l'utilisation de Docker Compose. Cet outil va **faciliter le déploiement d'un nouveau conteneur** grâce à un **fichier de configuration au format YAML**.

Autrement dit, vous importez le fichier "**docker-compose.yml**" dans Container Manager et vous validez pour lancer le téléchargement de l'image du conteneur, ainsi que la création et la configuration du conteneur en lui-même. Ceci vous évite de créer le conteneur pas-à-pas en suivant l'assistant de Synology. Désormais, j'ai pris l'habitude d'utiliser cette méthode pour déployer une nouvelle application conteneurisée sur un NAS Synology.

image.png

Il est important de préciser que cela **améliore la gestion et la portabilité des conteneurs Docker.** En effet, ce fichier de configuration contient toutes les informations relatives à l'exécution de ce conteneur : version de Docker nécessaire, image, mode du réseau, stockage, etc. De plus, **un projet peut correspondre à une application multi-conteneurs**.

Conteneur

La section "**Conteneur**" contient la liste de tous les conteneurs présents sur votre NAS Synology, ainsi que leur statut. C'est également ici que vous pouvez créer un nouveau conteneur à l'aide de l'assistant graphique de DSM, obtenir des détails sur un conteneur, modifier sa configuration, etc....

En passant par la section "**Conteneur**", vous effectuez **la création manuelle d'un conteneur** en faisant abstraction sur le fait de pouvoir utiliser Docker Compose (via la section "**Projet**"). Avant que l'application Docker devienne Container Manager, c'était la seule option.

image.png

Image

La section "**Image**" contient **la liste de toutes les images Docker présentes sur votre NAS Synology**. À chaque fois, plusieurs informations sont indiquées : nom de l'image, la version, la

taille de l'image et l'heure de création de l'image Docker (ce qui ne correspond pas à la date et l'heure à laquelle vous avez effectué le téléchargement).

La première colonne indique le statut : quand c'est bleu, c'est que l'image est utilisée, c'est-à-dire associée à un conteneur, alors que quand c'est blanc, elle n'est pas utilisée.

image.png

Registre

La section "**Registre**" donne accès à **la liste des images que vous pouvez télécharger sur votre NAS** et exploiter ensuite dans des conteneurs Docker. Par défaut, Container Manager s'appuie sur le dépôt officiel "**Docker Hub**", mais en cliquant sur le bouton "**Paramètres**", vous pouvez ajouter des dépôts privés. Le nombre d'étoiles indique la popularité de l'image, c'est donc un indicateur important.

image.png

Réseau

La section "**Réseau**", comme son nom l'indique, donne accès à la **gestion du réseau pour les conteneurs**. Par défaut, Docker sur Synology est accompagné par deux réseaux : "**host**" et "**bridge**", mais il est possible d'en créer d'autres.

En mode "**bridge**", les conteneurs peuvent **communiquer avec le réseau local sur lequel est connecté le NAS** tout en étant isolé. Tous les conteneurs connectés à un même réseau bridge peuvent également communiquer entre eux. Docker s'occupe de **faire le pont entre le réseau du conteneur et le réseau local**. Il s'agit du type de réseau par défaut.

Un conteneur connecté en mode "**host**" partage directement le réseau de l'hôte Docker, c'est-à-dire du NAS. Le conteneur utilise l'adresse IP de l'hôte directement. Il n'y a pas un réseau virtuel permettant d'interconnecter plusieurs conteneurs comme avec le mode "**bridge**". En complément, nous avons le pilote "**macvlan**" qui permet au conteneur d'avoir son adresse MAC ainsi que son adresse IP, et ainsi d'être visible sur le réseau local comme un hôte à part entière (cette configuration est utile dans certains cas, notamment si le conteneur hébergeur un serveur PXE).

image.png

Journal

La section "**Journal**" contient l'historique des actions effectuées via l'interface de "**Container Manager**" : téléchargement d'une image, création d'un conteneur, création d'un projet, démarrage ou arrêt d'un conteneur, etc... Il est possible de filtrer le journal par sévérité (Infos, avertissements, erreurs).

La gestion des données des conteneurs Docker

Chaque conteneur déployé a besoin de pouvoir stocker ses données. **Je vous recommande de créer un sous-dossier par conteneur dans le répertoire "docker" créé par Container Manager.** Par exemple, si vous souhaitez déployer "**Homer**" dans un conteneur, vous créez un répertoire "**homer**" sous "**docker**" et dans la configuration du conteneur Homer, il faudra pointer vers ce répertoire.

Voici un exemple :

image.png

Ceci vous permettra d'organiser données associées à vos conteneurs Docker déployés sur votre NAS Synology. Dans le répertoire du conteneur, il pourra y avoir d'autres dossiers (data, config, etc...) en fonction des besoins du conteneur en lui-même.

Créer un utilisateur dédié pour exécuter les conteneurs Docker

Lors de la configuration d'un conteneur Docker, notamment à partir d'une configuration Docker Compose, il est très fréquent de devoir préciser avec quel utilisateur nous souhaitons exécuter le conteneur.

Pour des raisons de sécurité, **évitez d'exécuter le conteneur avec un compte utilisateur qui est administrateur du NAS.** À la place, utilisez un **compte utilisateur dédié pour Docker**, par exemple, nommé "**docker"** et qui aura des permissions de lecture et écriture sur le répertoire "**docker**" et son contenu. C'est tout. Il n'aura pas accès aux autres applications, ni même à l'interface de DSM.

> Dans cet exemple, le compte "docker" sera membre du groupe "users" présent par défaut et nous lui refuserons l'accès à toutes les applications de façon explicite. Vous pouvez aussi créer un groupe "docker" et configurer les permissions sur le groupe, puis ajouter l'utilisateur "docker" uniquement à ce groupe à la place de "users".

> Toutes les images et les scénarios ne permettant pas de spécifier l'utilisateur avec lequel vous souhaitez exécuter le conteneur. En effet, cela dépend des privilèges requis par le conteneur.

Pour créer ce nouveau compte utilisateur, suivez la procédure suivante :

1 - Cliquez sur "**Panneau de configuration**" puis "**Utilisateur et groupe**".

2 - Cliquez sur le bouton "**Créer**" à partir de l'onglet "**Utilisateur**".

3 - Indiquez un nom, par exemple "**docker**", ainsi qu'une description et un mot de passe (que vous stockez dans votre coffre-fort de mots de passe). Cochez également l'option "**Ne pas autoriser l'utilisateur à changer le mot de passe du compte**".

image.png

4 - Ajoutez l'utilisateur au groupe "**users**" (ou à votre groupe "**docker**" si vous l'avez créé en amont).

image.png

5 - Attribuer les permissions de **lecture et écriture** sur le dossier partagé "**docker**" à cet utilisateur.

image.png

6 - Passez l'étape correspondante à la gestion du quota.

7 - Refusez l'accès, de façon explicite, à toutes les applications. Cet utilisateur n'a aucune raison d'avoir accès à l'interface DSM ou à d'autres fonctions.

image.png

8 - Poursuivez jusqu'à la fin pour créer l'utilisateur.

image.png

Voilà, le compte utilisateur pour Docker a été créé ! Il ne restera plus qu'à récupérer l'UID et le GID pour les spécifier dans un conteneur qui doit être exécuté avec ce compte.

Voici un exemple pour le mappage du répertoire de données dans un conteneur, ainsi que de l'utilisation de l'utilisateur "**docker**" pour exécuter un conteneur Docker :

image.png

Déployer un premier conteneur

Pour que vous puissiez faire vos premiers pas avec le déploiement d'un conteneur Docker sur un NAS Synology, nous allons faire simple et déployer un conteneur basé sur l'image "**httpd**". Ceci correspond à un serveur web **Apache2**.

Nous allons créer un répertoire nommé "**httpd**" dans "**docker**". Puis, dans le répertoire "**httpd**", nous allons créer le répertoire "**websites**" qui sera destiné à stocker les données de notre site web

statique. Notre conteneur sera basé sur l'image "**httpd**" visible sur l'image ci-dessous.

image.png

Désormais, nous allons pouvoir créer ce nouveau conteneur Apache2. Suivez les étapes suivantes :

1 - À partir de l'interface "**Container Manager**", cliquez sur "**Projet**" puis sur le bouton "**Créer**".

2 - Nommez ce projet "**apache2_httpd**" puis indiquez "**/docker/httpd**" comme chemin.

image.png

Vous devez également indiquer le contenu de votre fichier "**docker-compose.yml**". Pour déployer un serveur Apache à partir de l'image la plus récente ("**image: httpd:latest**") dans un conteneur nommé "**httpd-website**", voici le code à utiliser :

```
version: '3.9'
services:
  apache:
    image: httpd:latest
    container_name: httpd-website
    ports:
    - '9090:80'
    volumes:
    - /volume1/docker/httpd/websites:/usr/local/apache2/htdocs
```

Il y a également deux directives pour effectuer le mappage du port et du stockage :

- **Le conteneur sera accessible sur le port 9090 en externe, tandis que le port interne est 80 (http).**
- **Le répertoire "/usr/local/apache2/htdocs" du conteneur et correspondant à la racine du serveur Web, sera mappé avec le répertoire "/volume1/docker/httpd/websites" du NAS.**

> Pour obtenir le fichier de configuration Docker Compose d'une application, consultez la documentation officielle de l'application en question. En général, il y a des instructions puisque ce type de configuration tend à se démocratiser.

Vous pouvez continuer jusqu'à la fin et valider.

Container Manager va télécharger l'image associée à notre projet et construire le conteneur associé.

image.png

Une fois que ce sera terminé, le conteneur Apache2 sera actif et accessible. Le statut global du projet est visible dans "**Projet**", tandis que le statut du conteneur "**httpd-website**" est visible dans "**Conteneur**". Cette section permet d'afficher le statut par conteneur, car un projet peut regrouper plusieurs conteneurs. Dans la section "**Image**", celle correspondante au conteneur Apache2 a bien été téléchargée.

image.png

À partir de "**Projet**", si vous cliquez sur le nom du projet, vous pouvez obtenir des informations très intéressantes :

- **Conteneurs** : la liste des conteneurs associés à ce projet avec le statut correspondant
- **Statistiques** : les ressources consommées (CPU, RAM, réseau)
- **Configurations YAML** : le code du fichier docker-compose.yml de ce projet. Vous pouvez l'éditer quand le projet (et donc les conteneurs) est arrêté.
- **Paramètres** : publier ce conteneur via l'application Web Station de DSM. Pour avoir des paramètres avancés, il faut cliquer sur le conteneur via le menu "**Conteneur**", sans passer par "**Projet**".

image.png

Désormais, nous allons tenter d'accéder à la page d'accueil de notre serveur web Apache2.

À partir d'un navigateur, il est possible d'**accéder à notre serveur web Apache2.** Il suffit de saisir **l'adresse IP du NAS, suivie par le numéro de port 9090**. Actuellement, il n'y a aucun fichier, donc la page web affiche la directive "**Index of /**".

image.png

Dans le répertoire "websites" de votre NAS, vous pouvez déposer un ou plusieurs fichiers.

image.png

Par exemple, voici un fichier nommé "**index.html**" avec le code suivant :

```
<html>
<body>
<h1>Demo IT-Connect</h1>
</body>
</html>
```

Le fichier "**index.html**" a été copié dans le répertoire "**/docker/httpd/websites**". Désormais la page retournée est différente :

image.png

Notre conteneur Apache2 est opérationnel !

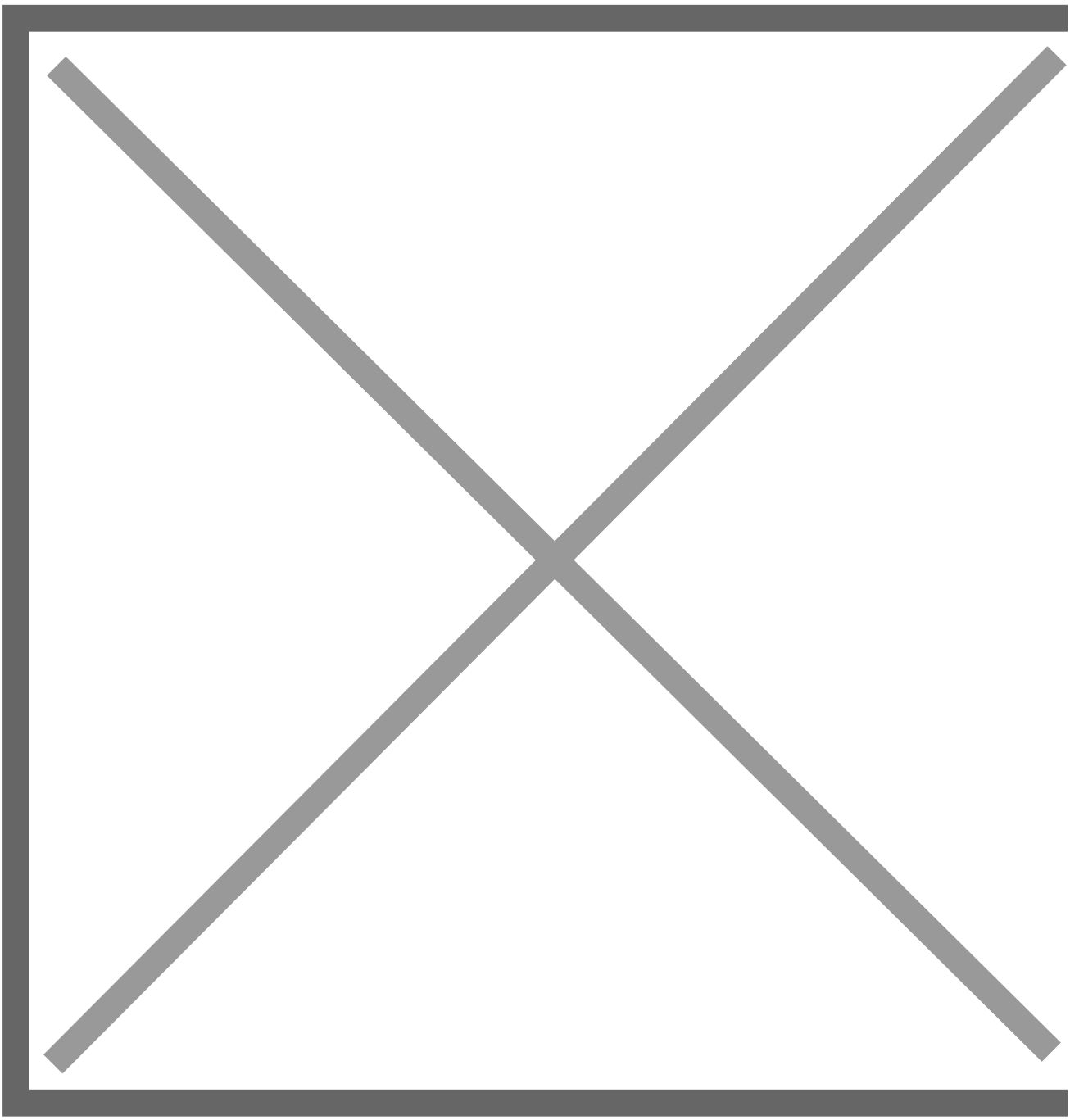
DocuSeal : la solution de signature électronique

Dans ce tutoriel, nous allons apprendre à installer DocuSeal sur un NAS Synology, à l'aide d'un conteneur Docker, via l'application Container Manager. Ceci va vous permettre d'héberger sur votre infrastructure une solution de signature électronique open source. Ainsi, vous gardez la maitrise de vos données, tout en permettant la mise en place d'un processus de signature électronique pour votre entreprise.

Déployer DocuSeal avec Docker

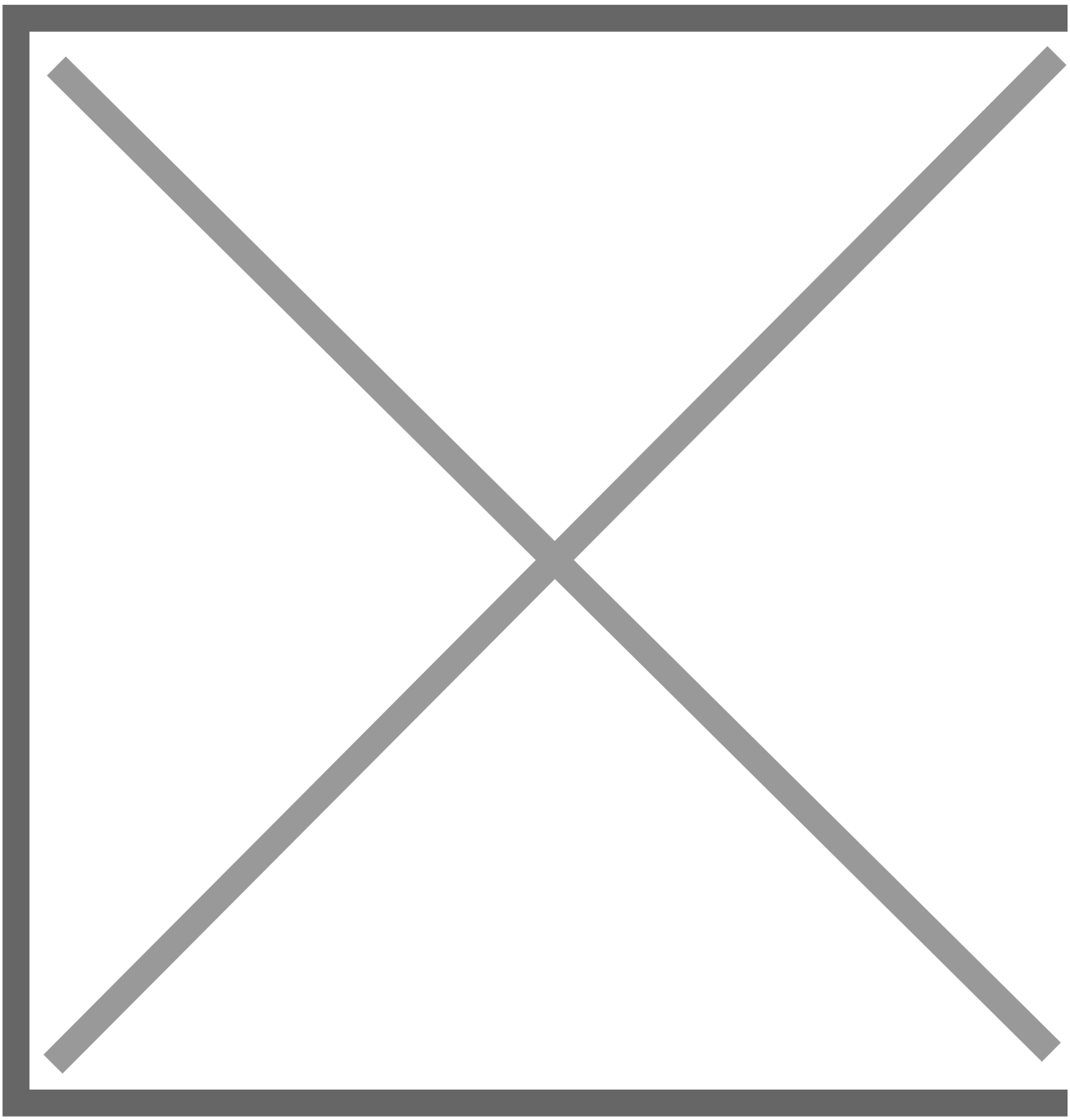
Avant toute chose, vous devez installer l'application "**Container Manager**" sur votre NAS, si ce n'est pas déjà fait. Puis, vous devez créer un dossier dédié pour ce nouveau projet basé sur 2 conteneurs.

Pour ma part, il s'agit du répertoire "**docuseal**" créé sous "**docker**". Il y a aussi deux sous-dossiers à créer : "**data**" et "**postgresql**" afin de stocker les données de l'application et la base de données. Ce qui donne le schéma suivant :



Ensuite, ouvrez "**Container Manager**" puis cliquez sur "**Projet**" afin de créer un nouveau projet en cliquant sur le bouton nommé "**Créer**".

Vous devez donner un nom à ce projet, sélectionner le répertoire créé précédemment et indiquer le code de configuration Docker Compose. Ce projet s'appuie sur 2 conteneurs : un conteneur pour l'application DocuSeal en elle-même et un conteneur PostgreSQL pour la base de données. Ce qui donne :



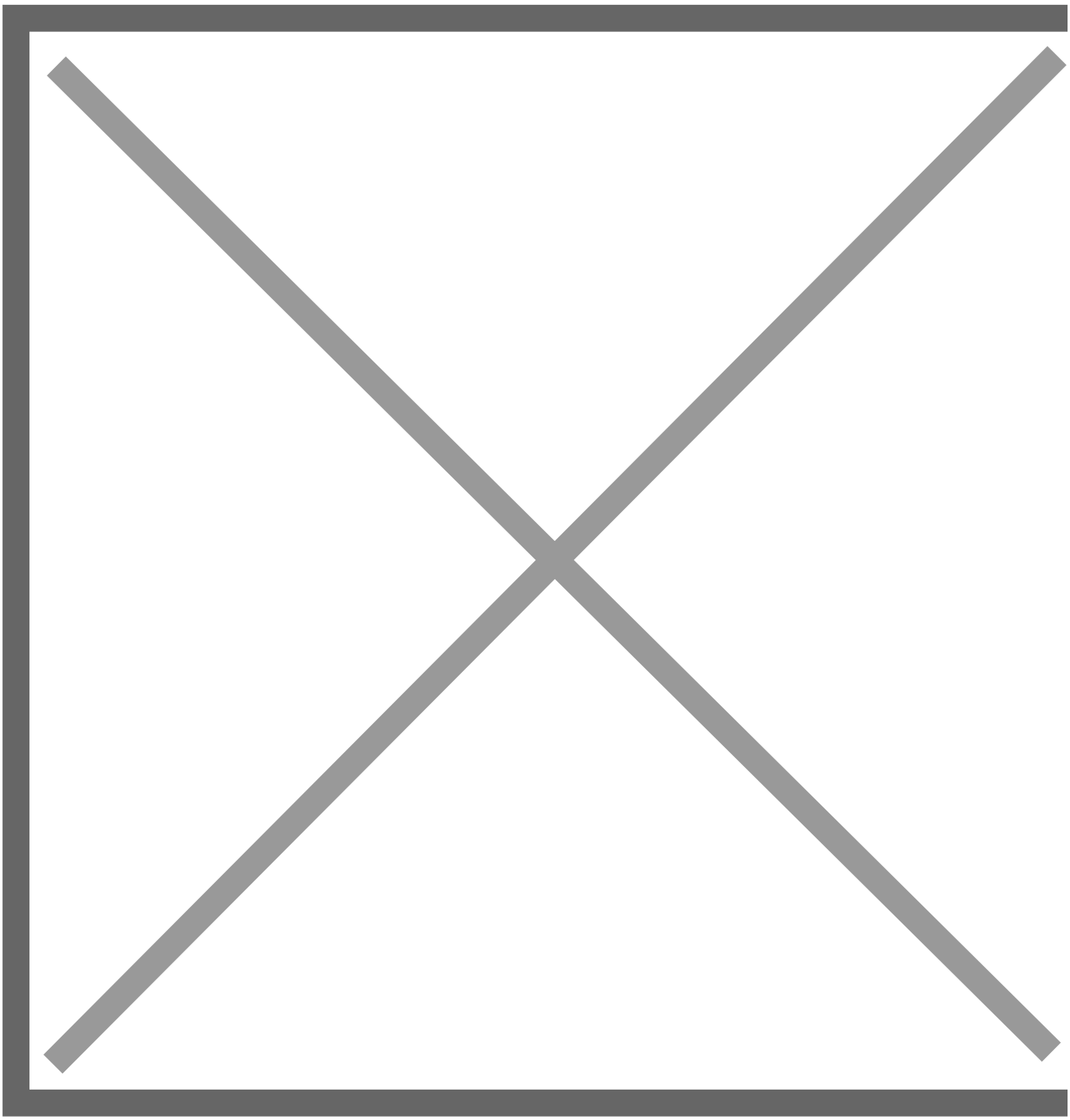
Voici le code Docker Compose complet :

```
services:
  app:
    depends_on:
      postgres:
        condition: service_healthy
    image: docuseal/docuseal:latest
    ports:
      - 3000:3000
    volumes:
```

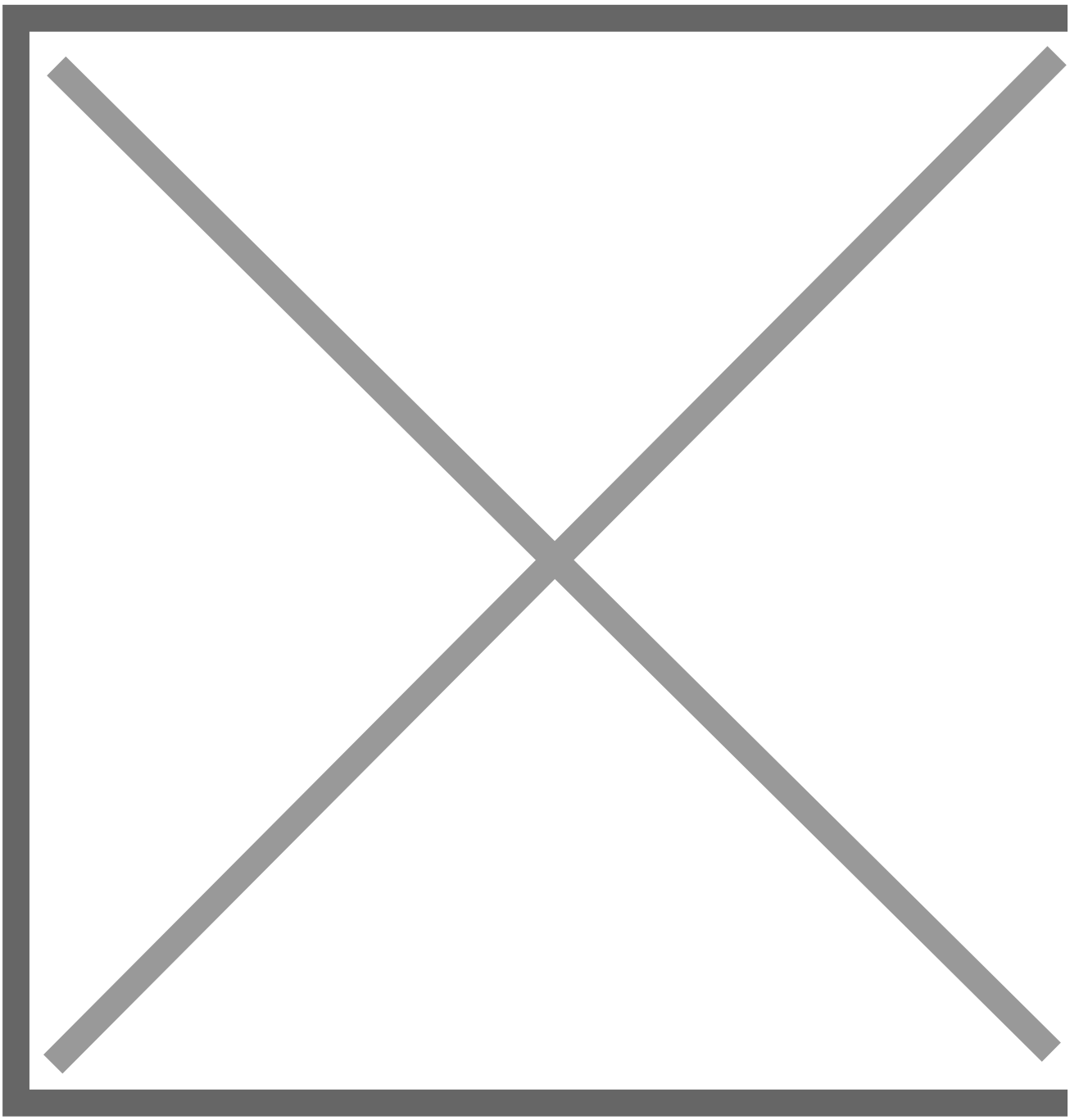
```
    - /volume1/docker/docuseal/data:/data/docuseal
  environment:
    - FORCE_SSL=${HOST}
    - DATABASE_URL=postgresql://postgres:MotDePasseBdd@postgres:5432/docuseal

postgres:
  image: postgres:15
  volumes:
    - /volume1/docker/docuseal/postgresql:/var/lib/postgresql/data
  environment:
    POSTGRES_USER: postgres
    POSTGRES_PASSWORD: MotDePasseBdd
    POSTGRES_DB: docuseal
  healthcheck:
    test: ["CMD-SHELL", "pg_isready -U postgres"]
    interval: 5s
    timeout: 5s
    retries: 5
```

Ici, le mot de passe de l'utilisateur dédié pour la base de donnée est "**MotDePasseBdd**". Pour le changer, modifier la directive "**POSTGRES_PASSWORD**", ainsi que le mot de passe présent dans la directive "**DATABASE_URL**". Il est à noter que l'application sera accessible sur le **port 3000**, mais vous pouvez modifier ce port si besoin (directives "**ports**"). Veillez aussi à mettre à jour les deux directives "**volumes**" pour que cela correspondent à votre arborescence de dossiers.

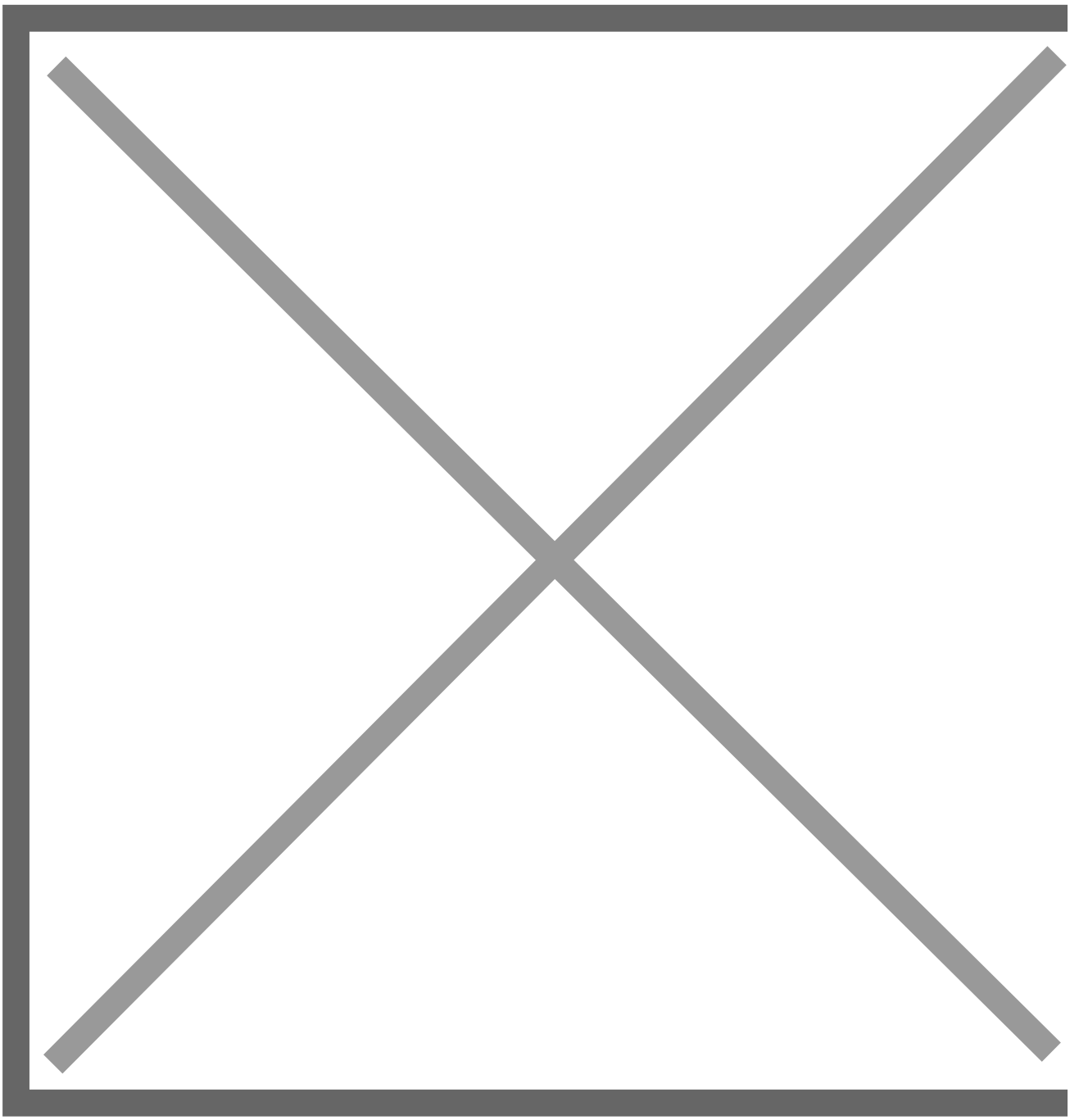
Ensuite, poursuivez jusqu'à la fin et lancez la création du projet... Patientez pendant le téléchargement des images Docker et la création des conteneurs associés.



Le projet a été créé et il est bien associé à deux conteneurs :



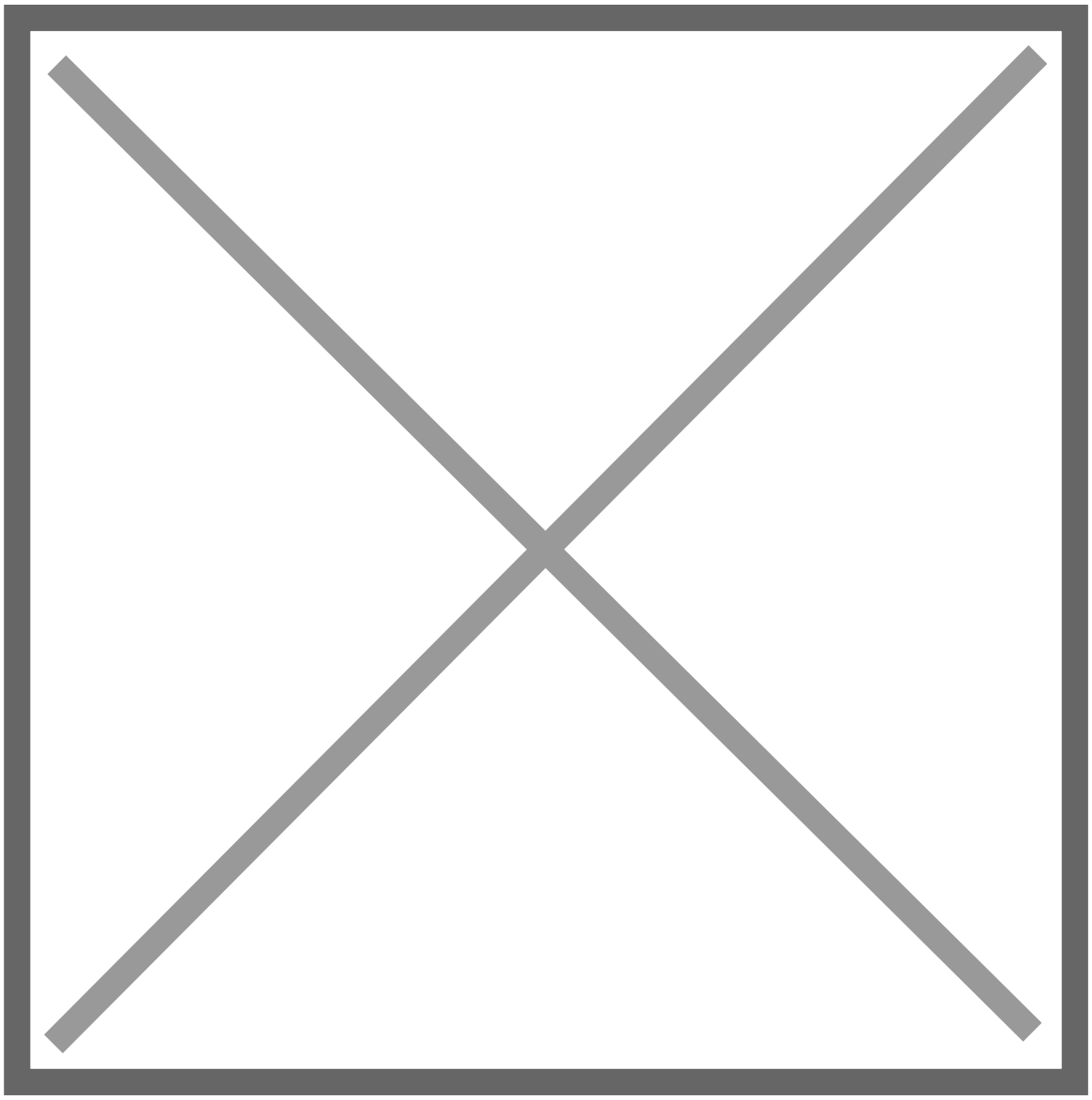
Si le pare-feu de votre NAS est activé, veillez à créer une règle de pare-feu pour autoriser les connexions sur le port 3000.



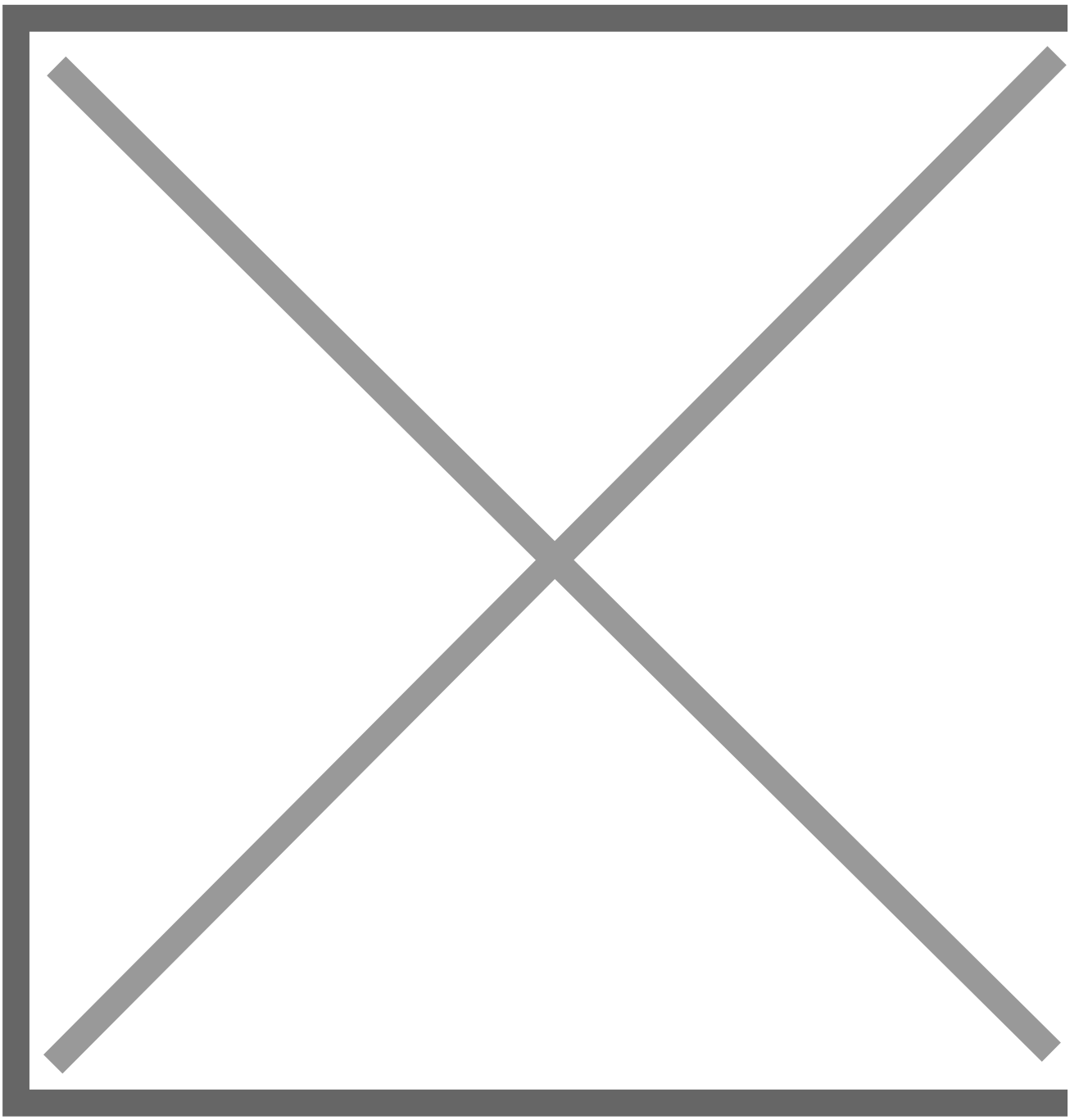
Ensuite, vous pouvez accéder à l'application DocuSeal de cette façon :

- **http://<adresse IP de votre NAS>:3000**

La première étape consiste à créer le compte administrateur de la plateforme... Renseignez le formulaire.



Voilà, vous avez accès à la solution DocuSeal ! Pour apprendre à l'utiliser, vous pouvez lire notre tutoriel de prise en main (lien en introduction).

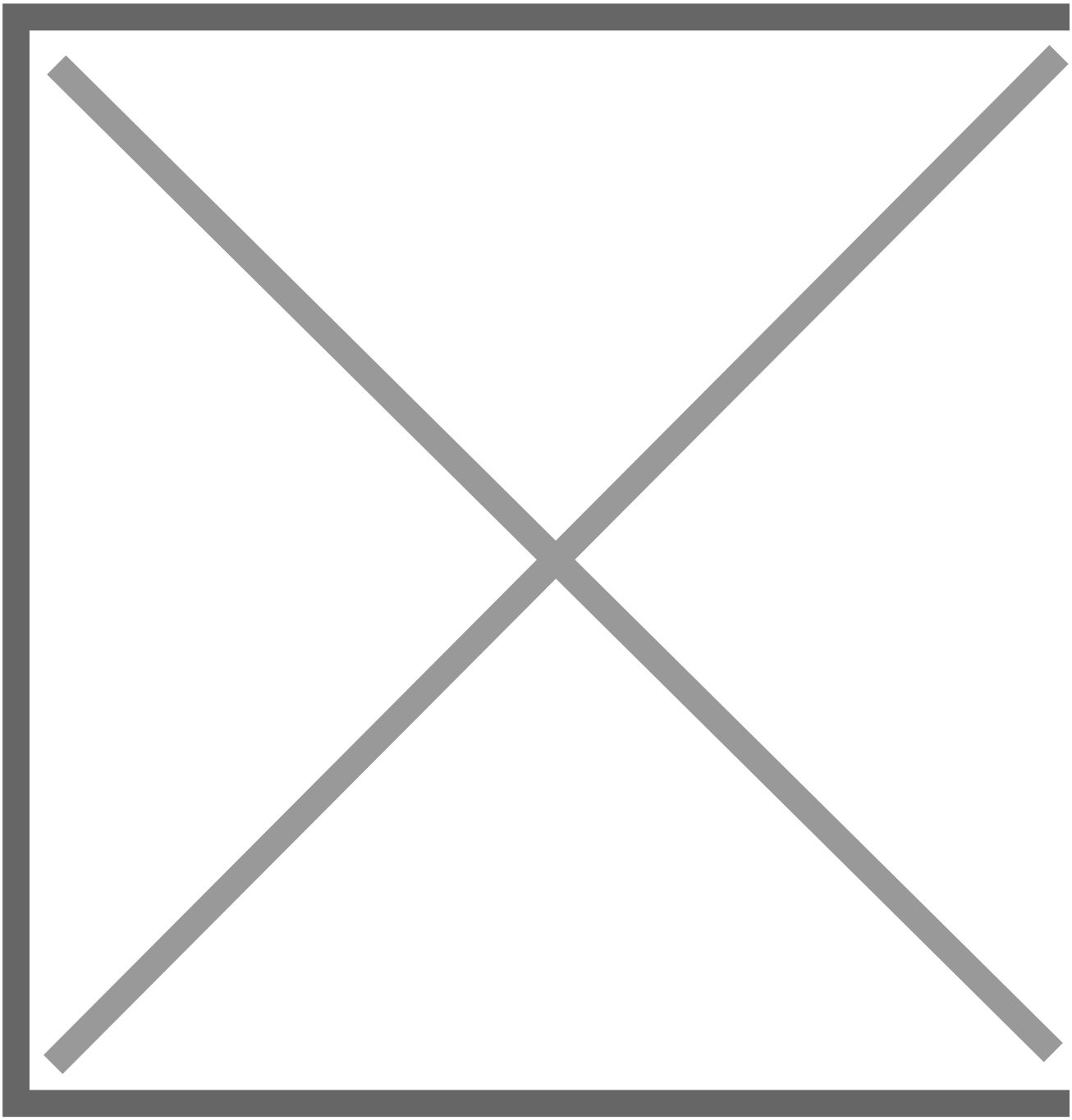


Publier DocuSeal avec un NAS Synology

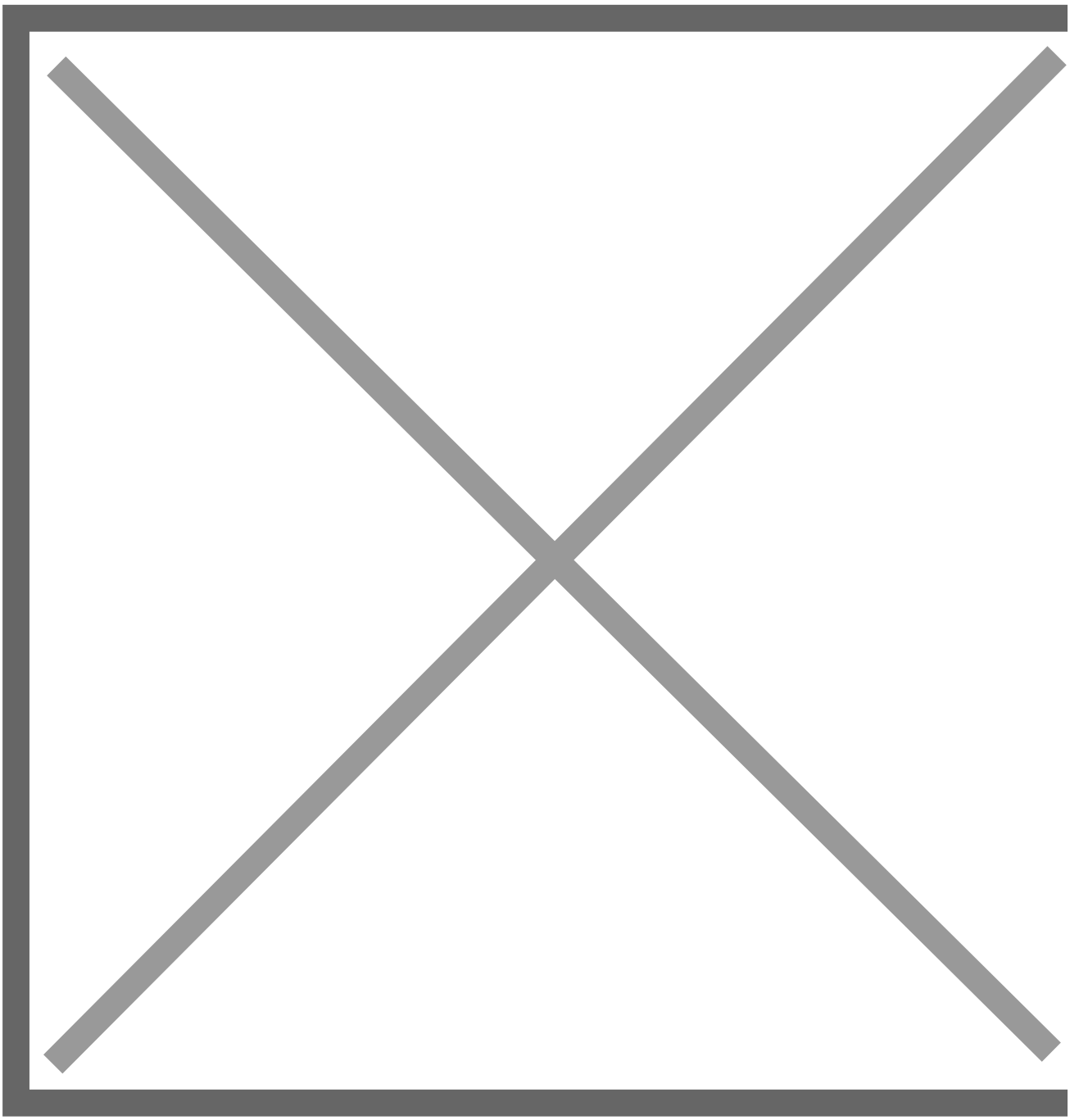
Pour que DocuSeal soit accessible depuis Internet, vous avez plusieurs options. L'une de ces options, c'est d'utiliser le reverse proxy de DSM couplé au DDNS afin de publier l'application et obtenir un certificat TLS via Let's Encrypt.

Si le NAS est associé au nom de domaine "**itconnect.synology.me**" et qu'il est accessible sur le **port 6001** (port de DSM), alors nous pouvons imaginer accéder à DocuSeal via une URL telle que : **https://docuseal.itconnect.synology.me:6001**, tout en ayant un conteneur en écoute sur le **port 3000**.

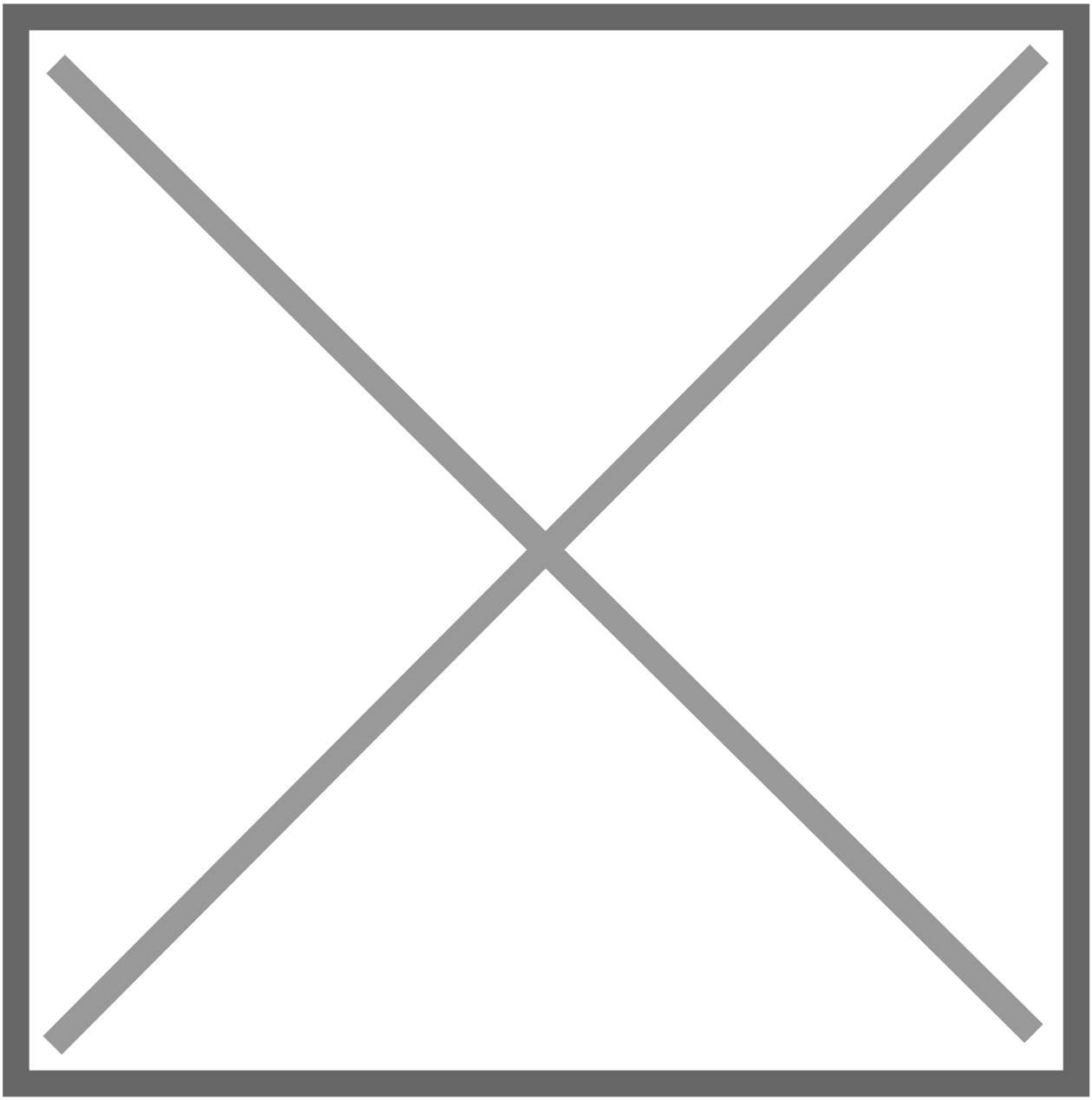
Ceci implique de configurer le reverse proxy de DSM de cette façon :



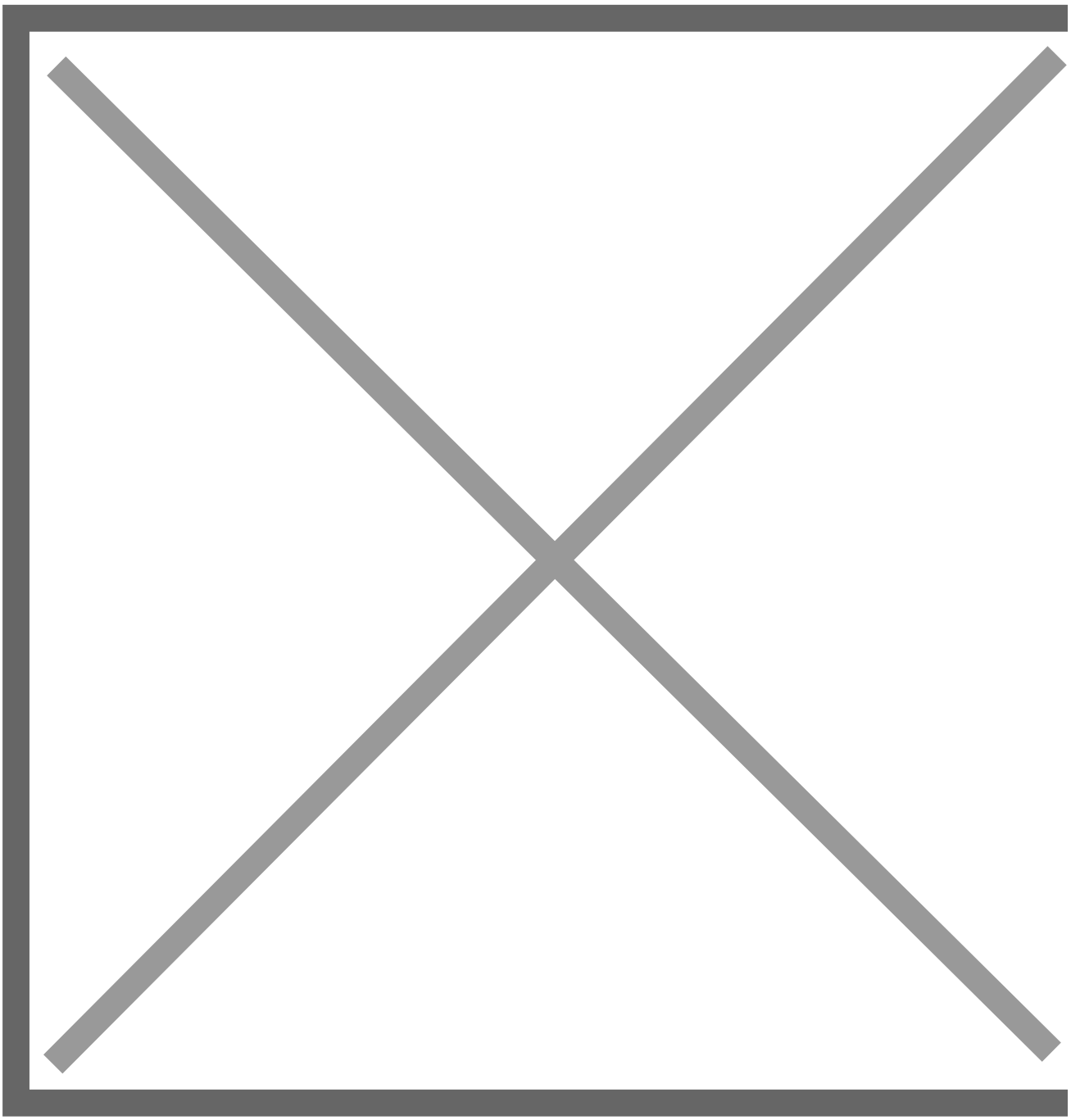
Ensuite, il sera nécessaire de demander l'obtention d'un certificat Let's Encrypt. Dans ce cas, il faut accéder à la gestion des certificats de DSM et demander un nouveau certificat Let's Encrypt pour "**docuseal.itconnect.synology.me**".



Vous devez remplir le formulaire de cette façon :



En complément, il sera nécessaire de configurer une règle de NAT sur votre routeur ou votre firewall pour permettre l'accès externe à l'application. Après avoir effectué cette configuration, vous pourrez accéder à l'application DocuSeal via ce nom de domaine personnalisé :



iVentoy : serveur PXE

Dans ce tutoriel, nous allons apprendre à déployer iVentoy dans un container Docker sur un NAS Synology, à l'aide de l'application Container Manager. Ceci va permettre d'avoir un serveur PXE sur notre NAS Synology ! iVentoy prend en charge des images ISO Windows, Linux, etc.

Créer un répertoire pour le conteneur iVentoy

Nous allons commencer par créer un répertoire dédié pour ce conteneur, ainsi que plusieurs sous-répertoires nécessaires pour transférer des données à l'application. Ainsi, dans le répertoire "**docker**", nous allons créer un répertoire nommé "**iventoy**". Au sein de celui-ci, trois autres répertoires sont à créer : "**data**", "**iso**", "**log**".

Ce qui donne l'arborescence suivante :

image.png

Copier les fichiers "data" d'iVentoy

Au sein du répertoire "**data**" que nous venons de créer, nous devons stocker deux fichiers : "**iventoy.dat**" et "**mac.db**". Sans cela, le conteneur ne fonctionnera pas. **Où faut-il récupérer ces fichiers ?** Bonne question ! Vous devez télécharger la dernière version d'iVentoy sur GitHub :

- GitHub - iVentoy

Dans l'archive obtenue, vous pourrez trouver un dossier nommé "**data**" avec ces deux fichiers. Il vous suffit de les envoyer vers le NAS. Comme ceci :

image.png

En complément, dans le dossier "**iso**", vous pouvez **copier-coller les images ISO des systèmes d'exploitation** que vous souhaitez **déployer par le réseau avec iVentoy**.

Créer le conteneur iVentoy

Désormais, nous allons pouvoir créer le conteneur iVentoy à partir de Container Manager. Cliquez sur "**Projet**" puis créez un nouveau projet.

Nommez ce projet "**iventoy**" et sélectionnez le chemin "**/docker/iventoy**" correspondant au répertoire racine créé précédemment. De plus, sélectionnez la valeur "**Créer un fichier docker-compose.yml**" pour l'option "**Source**" puisque nous allons utiliser une configuration Docker Compose.

image.png

Que faut-il indiquer ici ? Comme point de départ, nous allons utiliser la configuration Docker Compose disponible sur cette page GitHub. Mais, nous devons modifier cette configuration, car elle n'est pas adaptée pour Synology.

image.png

Nous devons **personnaliser cette configuration**, notamment car **iVentoy doit communiquer en direct avec notre réseau local**. En effet, comme il joue le rôle de serveur DHCP et serveur PXE, il doit être joignable par les hôtes du réseau.

Pour la couche réseau de ce conteneur, nous allons utiliser le **pilote macvlan de Docker** pour répondre à ce besoin : **notre conteneur iVentoy aura une adresse IP statique sur notre réseau local.** Un réseau nommé "**macvlan**" sera créé dans Container Manager.

Voici le code complet et, à la suite, des explications supplémentaires :

```
---
version: '3.9'
services:
  iventoy:
    image: ziggyds/iventoy:latest
    container_name: iventoy
    restart: always
    privileged: true #must be true
    ports:
      - 26000:26000
      - 16000:16000
      - 10809:10809
      - 67:67/udp
      - 69:69/udp
    volumes:
      - /volume1/docker/iventoy/iso:/app/iso
      - /volume1/docker/iventoy/data:/app/data
      - /volume1/docker/iventoy/log:/app/log
```

```
    environment:
      - AUTO_START_PXE=true # optional, true by default
    networks:
      macvlan:
        ipv4_address: 192.168.1.170
volumes:
  iso:
    external: true
  data:
    external: true
networks:
  macvlan:
    name: macvlan
    driver: macvlan
    driver_opts:
      parent: eth0
    ipam:
      config:
        - subnet: "192.168.1.0/24"
          ip_range: "192.168.1.192/28"
          gateway: "192.168.1.254"
```

Pour vous aider à comprendre cette configuration Docker Compose, voici quelques indications.

Ce conteneur doit être exécuté en mode privilégié, ce qui explique la présence de cette ligne :

```
    privileged: true #must be true
```

La section "**volumes**" sert à mapper les répertoires du conteneur avec ceux présents sur notre NAS. Le répertoire "**iso**" créé précédemment devra être utilisé pour stocker les images ISO que vous souhaitez utiliser dans iVentoy.

```
    volumes:
      - /volume1/docker/iventoy/iso:/app/iso
      - /volume1/docker/iventoy/data:/app/data
      - /volume1/docker/iventoy/log:/app/log
```

Le conteneur sera connecté au réseau "**macvlan**" et il aura l'adresse IP "**192.168.1.170**". Adaptez en fonction de votre réseau local.

```
    networks:
      macvlan:
        ipv4_address: 192.168.1.170
```

Cette partie de la configuration vise à créer le réseau "**macvlan**" dans Docker. Il est associé au sous-réseau "**192.168.1.0/24**", à la passerelle par défaut "**192.168.1.254**" . Nous attribuons aussi la plage d'adresses IP "**192.168.1.192/28**" (soit 14 adresses IP) au conteneur. Pour rappel, la version FREE d'iVentoy est, de toute façon, limitée à 20 adresses IP.

```
networks:
  macvlan:
    name: macvlan
    driver: macvlan
    driver_opts:
      parent: eth0
    ipam:
      config:
        - subnet: "192.168.1.0/24"
          ip_range: "192.168.1.192/28"
          gateway: "192.168.1.254"
```

Une fois que vous avez adapté cette configuration à votre environnement, poursuivez jusqu'à la fin pour créer le conteneur Docker.

image.png

Le conteneur iVentoy est actif, nous allons pouvoir tenter une connexion à l'interface web !

Accéder à l'interface web d'iVentoy

Pour accéder à l'interface de gestion d'iVentoy, nous devons **spécifier l'adresse IP du conteneur**, ainsi que le **numéro de port** (26000 pour la GUI Web).

http://192.168.1.170:26000/

Nous avons bien accès à l'interface d'administration. Si vous ne parvenez pas à accéder à l'interface d'iVentoy, vérifiez la configuration de votre pare-feu Synology, ainsi que votre fichier Docker Compose.

image.png

Container Manager

Stirling PDF : La boite à outils PDF

Dans ce tutoriel, nous allons apprendre à installer et à configurer la solution Stirling PDF à l'aide d'un conteneur Docker. Stirling PDF est une puissante boite à outils gratuite et open source pour gérer et manipuler vos fichiers PDF localement.

Accessible à partir d'un navigateur Web, cette solution va vous permettre d'effectuer des actions diverses et variées sur vos fichiers PDF, tout en gardant la maitrise de vos données. En effet, avec Stirling PDF, tout fonctionne entièrement sur votre machine locale, garantissant la confidentialité et le contrôle de vos données. Ceci pourra éviter que vos utilisateurs s'appuient sur des outils en ligne, au risque qu'il y ait une fuite de données par la même occasion.

Voici quelques-unes des fonctionnalités disponibles :

- **Fusionner** ou **diviser** des documents PDF
- **Extraire des pages** d'un document PDF
- **Réorganiser** ou **pivoter** les pages d'un document PDF
- **Convertir** des fichiers sources en PDF (image vers PDF, HTML vers PDF, Markdown vers PDF, etc.)
- **Convertir** un fichier PDF dans un autre format (PDF vers image, PDF vers Word, PDF vers HTML, etc.)
- **Compresser** un document PDF
- **Extraire** les **images** ou **ajouter** une **image**
- **Modifier** les **métadonnées** d'un document PDF
- **Signer** un document PDF (avec un certificat numérique ou une image de signature)
- **Ajouter** ou **supprimer le mot de passe** d'un PDF
- **Ajouter** un **filigrane** sur un document PDF
- **Visionneuse** de documents PDF
- Reconnaissance de caractères (**OCR**)
- Etc...

Toutes ces fonctionnalités sont gratuites puisque cette application est libre et open source. Elle intègre de nombreuses fonctionnalités dont certaines parfois réservées à des outils premium (payants).

image.png

Stirling PDF peut être installé sur une machine Windows, en local, car des exécutables sont proposés sur le GitHub officiel. Ceci implique l'**installation de Java sur la machine**. Pour ma part, je préfère m'orienter vers **un déploiement dans un conteneur Docker**, sur un **NAS Synology**, afin de **mettre à disposition l'application à X utilisateurs**. Ceci est d'autant plus pertinent que l'application est **accessible à partir d'un navigateur**, via l'URL de votre choix.

Installer Stirling PDF sur un NAS Synology

Pour effectuer l'installation sur un NAS Synology, nous allons utiliser l'application "**Container Manager**" (Docker) afin de pouvoir utiliser un fichier de configuration **Docker Compose**. Ceci vous permet d'utiliser cette configuration facilement pour déployer le conteneur sur d'autres plateformes.

Tout d'abord, nous allons créer l'arborescence de dossiers pour accueillir les données de Stirling PDF. Sous le répertoire "docker", voici les répertoires à créer :

- **stirling-pdf**
  - **trainingData**
  - **extraConfigs**

Le répertoire "**trainingData**" est utile uniquement si vous envisagez d'utiliser la fonction liée à l'OCR.

Ce qui donne :

image.png

Ensuite, lancez l'application "**Container Manager**" (Docker) sur votre NAS Synology. Créez un nouveau projet :

- **Projet > Créer**

Commencez par donner un nom au projet, par exemple "**stirling-pdf**". Puis, indiquez le chemin correspondant au dossier précédemment créé, à savoir "**/docker/stirling-pdf**".

image.png

Puis, sélectionnez "**Créer un fichier docker-compose.yml**" afin de pouvoir personnaliser le déploiement de ce projet basé sur l'image Docker "**frooodle/s-pdf**" dans sa dernière version (vis-à-vis du tag "**latest**"). Voici un aperçu du code de configuration Docker Compose :

image.png

Pour cet exemple, je vais utiliser la configuration Docker Compose suivante :

```
version: '3.3'
services:
  stirling-pdf:
    image: frooodle/s-pdf:latest
    ports:
      - '8080:8080'
    volumes:
      - /volume1/docker/stirling-pdf/trainingData:/usr/share/tessdata #Required for extra OCR languages
      - /volume1/docker/stirling-pdf/extraConfigs:/configs
#      - /location/of/customFiles:/customFiles/
#      - /location/of/logs:/logs/
    environment:
      - DOCKER_ENABLE_SECURITY=true
      - SECURITY_ENABLE_LOGIN=true
      - SECURITY_INITIALLOGIN_USERNAME=pdf
      - SECURITY_INITIALLOGIN_PASSWORD=IT-Connect
      - INSTALL_BOOK_AND_ADVANCED_HTML_OPS=false
      - LANGS=fr_FR
```

Quelques explications s'imposent :

- Le conteneur sera accessible sur le port "**8080**" puisqu'il est mappé sur le port "**8080:8080**" (port externe côté NAS : port interne dans le conteneur). Vous pouvez adapter cette valeur (premier numéro de port) si besoin, parce qu'un seul conteneur peut occuper chaque port.
- Sous "**volumes**", indiquez les chemins vers les répertoires "**trainingData**" et "**extraConfigs**" créé précédemment. Les répertoires "**customFiles**" et "**Logs**" sont facultatifs. Ici, ces deux lignes sont commentées.
- La directive "**SECURITY_ENABLE_LOGIN=true**" sert à activer la page de connexion sur Stirling PDF. Autrement dit, il conviendra de s'authentifier avant d'accéder à l'application. Ceci implique aussi de configurer la directive "**DOCKER_ENABLE_SECURITY=true**".
- La directive "**SECURITY_INITIALLOGIN_USERNAME=pdf**" sert à créer un utilisateur par défaut nommé "**pdf**"
- La directive "**SECURITY_INITIALLOGIN_PASSWORD=IT-Connect**" sert à attribuer le mot de passe "**IT-Connect**" à l'utilisateur par défaut.
- La directive "**INSTALL_BOOK_AND_ADVANCED_HTML_OPS=false**" sert à indiquer qu'il ne faut pas télécharger l'application Calibre sur Stirling PDF. Elle permet la conversion de PDF vers/depuis un livre et la conversion HTML avancée. Activez cette option si besoin.
- La directive "**LANGS=fr_FR**" sert à précise la langue pour la bibliothèque de polices personnalisées à installer pour la conversion de documents. Dans tous les cas, l'interface de l'application sera disponible en plusieurs langues, dont le français.

Une fois que c'est fait, poursuivez jusqu'à la fin de l'assistant pour lancer la création du projet. L'image Docker va être téléchargée et le conteneur configuré puis exécuté.

image.png

L'application Stirling PDF est désormais exécutée au sein du conteneur Docker. Stirling PDF est une application assez gourmande en RAM, probablement à cause de son lien étroit avec Java. Le conteneur consomme entre 400 et 700 Mo de RAM, ce qui n'est pas neutre.

image.png

Découverte de Stirling PDF

Première connexion

L'accès à l'application s'effectue à partir d'un navigateur Web. Il suffit de préciser l'adresse IP ou le nom de domaine du NAS, suivi du port "8080", comme ceci :

```
http://192.168.1.200:8080/
```

Si vous avez activé l'authentification, vous devez vous connecter avec le compte créé par défaut (selon les informations définies dans le Docker Compose).

image.png

Voilà, vous êtes connecté à Stirling PDF ! Vous pouvez profiter de l'ensemble des outils intégrés à cette fabuleuse boite à outils !

image.png

Gestion des utilisateurs

Avant d'évoquer les outils en eux-mêmes, je vous recommande de changer le mot de passe du compte utilisateur créé par défaut. Vous pouvez aussi créer d'autres utilisateurs. Pour accéder à la gestion de votre compte, cliquez sur "**Paramètres**" en haut à droite (icône en forme de roue crantée) puis cliquez sur "**Paramètres du compte**".

Ici, vous pourrez changer le nom d'utilisateur et le mot de passe du compte.

image.png

Si vous descendez tout en bas de la page, vous pourrez cliquer sur le bouton "**Paramètres d'administration - Voir et ajouter des utilisateurs**". Ceci vous permet de **créer d'autres comptes utilisateurs**, avec différents niveaux de permissions : **administrateur**, **utilisateur**, etc.

image.png

Libre à vous de créer un ou plusieurs comptes. L'intérêt étant principalement de gérer l'accès à l'application. À ma connaissance, Stirling PDF n'a pas vocation à permettre à chaque utilisateur de gérer une bibliothèque de documents PDF.

À tout moment, vous pourrez créer, modifier ou supprimer des utilisateurs.

Utilisation des outils

Nous n'allons pas passer en revue l'ensemble des outils, car l'interface est simple d'utilisation et il y a énormément de possibilités. Il vous suffit de choisir l'outil de votre choix à partir du menu principal, ou de la page d'accueil. Vous pouvez aussi utiliser la fonction de recherche pour gagner du temps, ainsi que mettre certains outils en favoris.

image.png

La fonction "**Fusionner plusieurs PDF**" vous permet, comme son nom l'indique, de fusionner (concaténer) plusieurs documents PDF en un seul fichier.

image.png

Il y a également un outil multifonction qui donne accès à quelques fonctions basiques dans une même interface. Vous chargez un fichier, et ensuite, vous pouvez intervenir des pages, effectuer des rotations de pages, etc... Selon vos besoins.

image.png

À vous d'explorer les différents outils au fur et à mesure que les besoins se présenteront.

Aller plus loin dans la configuration

Est-ce qu'il y a des paramètres de configuration supplémentaires ? La réponse est oui. La configuration de Stirling PDF s'appuie sur un fichier au format YAML nommé "**settings.yml**". Il est situé à cet emplacement :

```
/docker/stirling-pdf/extraConfigs/settings.yml
```

Il permet de configurer l'application plus en profondeur. Nous pouvons **définir un nom personnalisé pour l'application**, mais aussi **configurer l'authentification OAuth2**. Ceci peut s'avérer utile pour s'appuyer sur **un service tiers pour l'authentification des utilisateurs** (Google, GitHub, KeyCloak).

Nous constatons aussi la présence de deux options ayant pour objectif de **protéger l'interface de connexion des attaques par brute force**. En effet, il y a un **verrouillage de comptes** activé par défaut : 5 mauvaises tentatives d'authentification à suivre vont engendrer le verrouillage d'un compte pendant 2 heures.

```
loginAttemptCount: 5 # lock user account after 5 tries
loginResetTimeMinutes: 120 # lock account for 2 hours after x attempts
```

Voici un aperçu de ce fichier de configuration que vous pouvez modifier selon vos besoins :

image.png

Pour aller plus loin, vous pouvez aussi publier l'application sur un nom de domaine à l'aide du reverse proxy de DSM.

NetAlertX : surveillance du réseau

Dans ce tutoriel, nous allons apprendre à installer l'application NetAlertX (Pi.Alert) sur un NAS Synology. L'installation sera effectuée dans un conteneur Docker, via Container Manager.

NetAlertX est une application gratuite et open source dont l'objectif est de surveiller votre réseau local dans le but de référencer tous les appareils connectés. Que ce soit un **appareil connu ou inconnu** (intrusion ?), il sera détecté par NetAlertX. Ainsi, vous pouvez **recevoir une notification** lorsqu'**un nouvel appareil se connecte pour la première fois** ou **si un appareil tombe en panne** (connecté puis déconnecté).

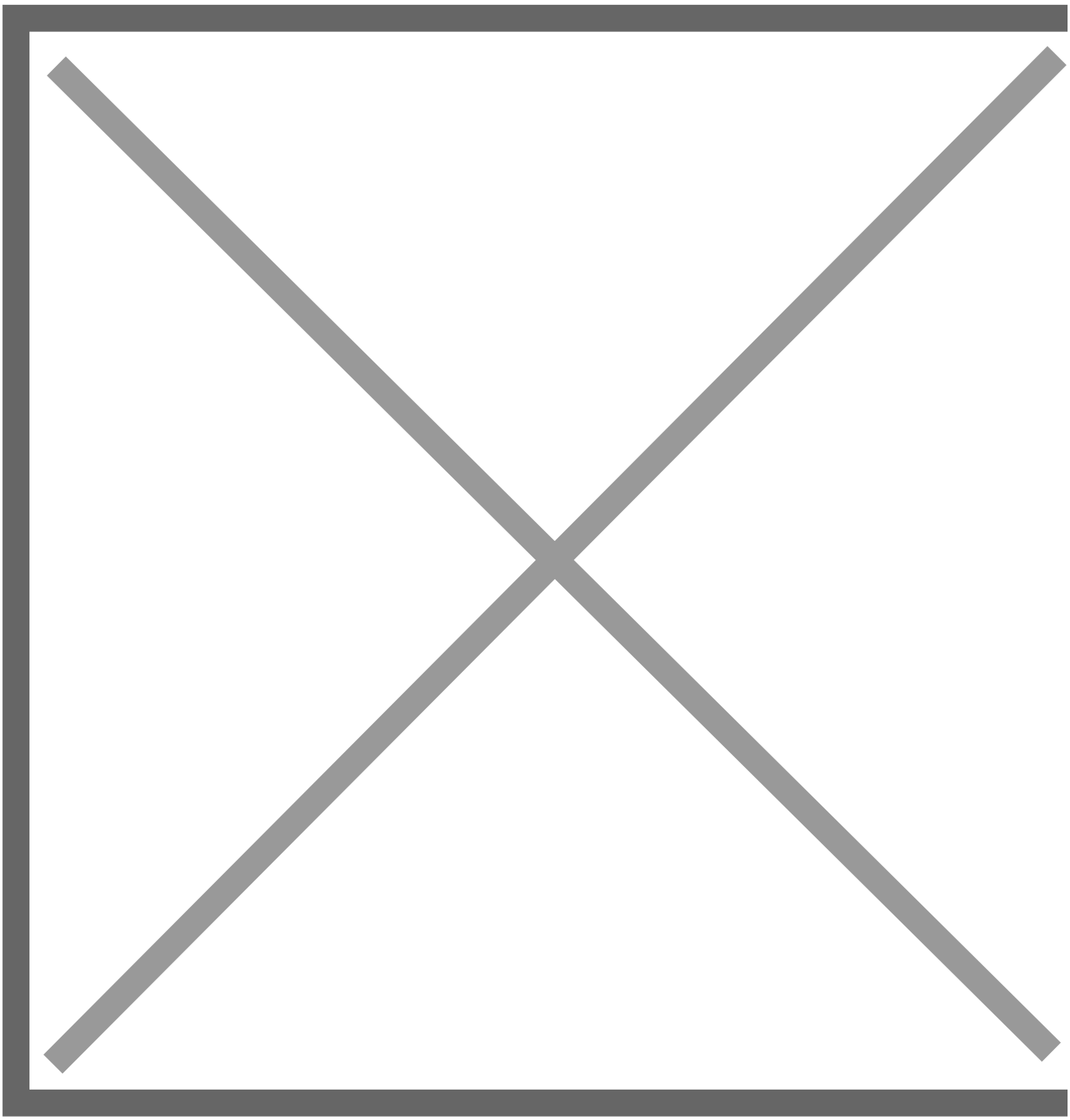
Déployer NetAlertX avec Docker

Avant toute chose, vous devez installer l'application "**Container Manager**" sur votre NAS, si ce n'est pas déjà fait. Puis, vous devez créer un dossier pour ce conteneur. Pour ma part, il s'agit du répertoire "**netalertx**" créé sous "**docker**".

Dans ce répertoire nouvellement créé, vous devez créer les trois sous-répertoires suivants :

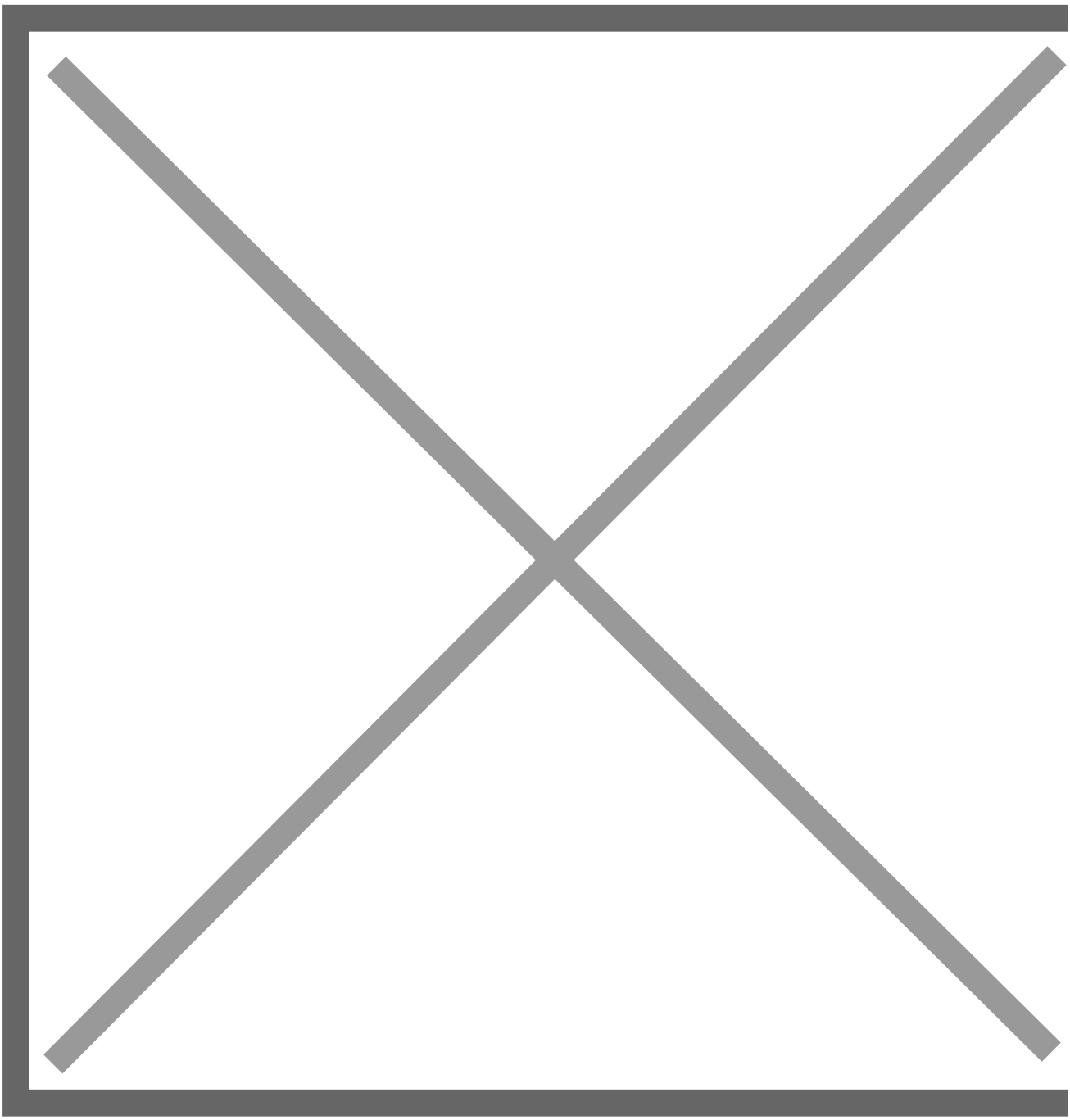
- **log**
- **db**
- **config**

Ils seront utilisés par l'application pour stocker des données (journaux, configuration, base de données).



Ensuite, ouvrez "**Container Manager**" puis cliquez sur "**Projet**" afin de créer un nouveau projet en cliquant sur le bouton nommé "**Créer**".

Vous devez donner un nom à ce projet, sélectionner le répertoire créé précédemment et indiquer le code de configuration Docker Compose permettant de déployer l'image Docker de l'application. Ce qui donne :



Voici le code Docker Compose (*aidez-vous des commentaires dans le fichier pour la compréhension de la config*) :

```
services:
  netalertx:
    image: jokobsk/netalertx:latest   # Nom de l'image Docker à utiliser
    container_name: netalertx         # Nom du conteneur Docker
    healthcheck:                      # Vérifier état de santé du conteneur (attention au port)
     test: curl -f http://127.0.0.1:15000/ || exit 1
    mem_limit: 1g                     # Limite RAM de 1 Go pour le conteneur
    security_opt:
```

```
 - no-new-privileges:true         # Option de sécurité contre l'élévation de privilèges
volumes:
 - /volume1/docker/netalertx/config:/app/config:rw          # Volume pour "config" (configuration)
 - /volume1/docker/netalertx/db:/app/db:rw                  # Volume pour "db" (base de données)
 - /volume1/docker/netalertx/log:/app/front/log:rw          # Volume pour "log" (journaux)
environment:
 TZ: Europe/Paris                # Fuseau horaire, ici Paris
 PORT: 15000                     # Port à utiliser sur la machine locale (adapter le healthcheck en fonction)
 HOST_USER_ID: 1032              # UserID à utiliser (n'utilisez pas "root")
 HOST_USER_GID: 100              # UserID à utiliser (n'utilisez pas "root")
 # ALWAYS_FRESH_INSTALL: true    # Réinitialise l'application (effacer toutes les données - désactivé ici)
network_mode: host               # Mode "host" obligatoire pour cette application
restart: on-failure:5            # Restart en cas de problème avec le conteneur ; 5 tentatives pour relancer
```

Docker Compose NetAlertX pour Synology.jpg

Dans le cas présent, le port sur lequel sera joignable l'application, est le **port 15000**. Veillez à vérifier également les chemins vers les dossiers, sous l'instruction "**volumes**".

Le **conteneur sera exécuté à partir de l'utilisateur "docker"**, créé par mes soins sur le NAS. Il s'agit d'un utilisateur non privilégié qui ne dispose d'aucune permission spéciale sur le NAS, si ce n'est le droit de lecture et écriture dans le répertoire "**docker**".

Quand la configuration est prête, poursuivez jusqu'à la fin et lancez la création du projet... Patientez pendant le téléchargement de l'image Docker et la création du conteneur associé.

Dès à présent, vous pouvez accéder à l'application NetAlertX de cette façon :

- **http://<adresse IP de votre NAS>:15000**

Vous avez désormais accès à l'application NetAlertX grâce à votre NAS !

NetAlertX sur NAS Synology avec Docker

Uptime Kuma : Surveiller vos sites web et conteneurs Docker

Dans ce tutoriel, nous allons apprendre à installer l'application Uptime Kuma sur un NAS Synology. L'installation sera effectuée dans un conteneur Docker, à partir du paquet "Container Manager" développé et maintenu par Synology.

Uptime Kuma est une solution de supervision simple et auto-hébergeable que vous pouvez utiliser pour surveiller vos sites web et conteneurs Docker. Cette solution se distingue par sa simplicité d'utilisation et son efficacité pour surveiller l'état en ligne de services web (blog, application métier, service SaaS, etc.). Vous pourriez même l'utiliser pour surveiller l'état de votre Home Lab.

Avec une interface moderne et intuitive, **Uptime Kuma permet de surveiller divers services via des protocoles tels que HTTP, HTTPS, TCP, ou encore des serveurs de jeux Steam**. Il propose également des fonctionnalités avancées, comme la surveillance des certificats TLS et l'envoi d'alertes via plus de 90 services (Telegram, Discord, Slack, etc.).

Déployer Uptime Kuma avec Docker

Avant toute chose, vous devez installer l'application "**Container Manager**" sur votre NAS, si ce n'est pas déjà fait. Puis, vous devez créer un dossier pour ce conteneur. Pour ma part, il s'agit du répertoire "**uptime-kuma**" créé sous "**docker**".

Dans ce répertoire nouvellement créé, vous devez créer un sous-dossier nommé "**data**" qui sera monté dans le conteneur. Nous obtenons le résultat suivant :

image.png

Ensuite, ouvrez "**Container Manager**" puis cliquez sur "**Projet**" afin de créer un nouveau projet en cliquant sur le bouton nommé "**Créer**".

Vous devez donner un nom à ce projet, sélectionner le répertoire créé précédemment, à savoir "**/docker/uptime-kuma**" et indiquer le code de configuration Docker Compose. Ce qui donne :

image.png

Voici le code Docker Compose :

```
services:
  uptime-kuma:
    image: louislam/uptime-kuma:1
    container_name: uptime-kuma
    volumes:
      - /volume1/docker/uptime-kuma/data:/app/data
    ports:
      - 3001:3001
    restart: always
```

Dans le cas présent, **l'application sera joignable en HTTP sur le port 3001**. Veillez à vérifier également le chemin vers le dossier "**data**", sous l'instruction "**volumes**".

Quand la configuration est prête, poursuivez jusqu'à la fin et lancez la création du projet... Patientez pendant le téléchargement de l'image Docker et la création du conteneur associé. Ceci peut nécessiter plusieurs minutes, en fonction de votre débit Internet.

Dès à présent, vous pouvez accéder à l'application Uptime Kuma de cette façon :

- **http://<adresse IP de votre NAS>:3001**

> Remarque : si le pare-feu de votre NAS est actif et que sa configuration est stricte, vous devez créer une règle pour autoriser les connexions sur le port 3001.

Vous avez désormais accès à l'application Uptime Kuma hébergée sur votre NAS ! Il ne reste plus qu'à passer à la phase de configuration.

image.png

Vaultwarden : Gestionnaire de mot de passe

Dans ce tutoriel, nous allons apprendre à déployer un serveur Vaultwarden sur un NAS Synology afin d'héberger notre propre solution de gestion de mots de passe ! Vous connaissez déjà surement Vaultwarden, car il s'agit d'une version non officielle de Bitwarden destinée à l'auto-hébergement !

Le gestionnaire de mots de passe Bitwarden peut être utilisé en mode SaaS, c'est-à-dire hébergé sur les serveurs de l'éditeur dans le Cloud, mais il peut aussi être auto-hébergé sur son propre serveur.

Par contre, il est considéré comme étant assez lourd, ce qui n'est pas très pratique si l'on veut utiliser un NAS ou pourquoi pas un Raspberry Pi.

La bonne nouvelle, c'est qu'il existe **Vaultwarden**, une **solution libre codée en Rust**, **beaucoup plus légère** et que l'on peut **héberger soi-même**. Avant, elle s'appelait "Bitwarden_RS" mais il y a eu un changement de nom pour une raison compréhensible : "*Ce projet était connu sous le nom de Bitwarden_RS et a été renommé pour se séparer du serveur officiel de Bitwarden dans l'espoir d'éviter toute confusion et tout problème de marque.*" - Vaultwarden est totalement adapté pour un NAS, un serveur en ligne (par exemple : un VPS), un Raspberry Pi ou un serveur en local.

Vaultwarden est une solution fiable et sécurisée qui reprend les fonctionnalités clés de Bitwarden, à savoir :

- Solution de type coffre-fort numérique pour stocker vos identifiants et mots de passe, vos informations de carte de paiement, des notes, etc...
- Gestion de plusieurs utilisateurs sur un même serveur (via un système d'inscription) et gestion d'organisations pour partager les identifiants entre utilisateurs
- Accès à votre coffre-fort sur tous les appareils associés à votre compte.
- Sécurité des données grâce au chiffrement bout en bout (seul l'utilisateur peut accéder à ses données)
- Générateur de mots de passe et de passphrases pour vous faciliter la création de mots de passe robustes
- Authentification à deux facteurs pour renforcer l'accès aux comptes
- Partage de fichiers ou de texte à l'aide de la fonctionnalité Send

Plutôt sympa pour **garder la maitrise de son coffre-fort de mots de passe**, tout en ayant accès à une solution moderne accessible via un navigateur, des extensions pour navigateur et une application mobile. D'ailleurs, **Vaultwarden est compatible avec les applications et**

extensions Bitwarden.

Pour **installer Vaultwarden sur un NAS Synology**, nous allons utiliser **un container Docker via l'application Container Manager de DSM**. Ce sera stable et facile à déployer, mais il conviendra de sauvegarder les données du container compte tenu de leur criticité. Il est à noter que l'application Docker s'appelle Container Manager depuis DSM 7.2.

Créer un container Docker "Vaultwarden"

Le NAS étant équipé de Container Manager, vous allez pouvoir **créer un nouveau container Vaultwarden** (en lieu et place de *Bitwarden_rs*).

Ouvrez Container Manager, cliquez à gauche sur "**Registre**" (1), recherchez "**vaultwarden**" en haut à droite (2), puis sélectionnez "**vaultwarden/server**" dans la liste (3) et cliquez sur "**Télécharger**" (4) pour que l'image de ce container soit téléchargée sur le NAS.

image.png

Au moment où vous cliquez sur "**Télécharger**", la fenêtre "**Choisir une identification**" apparaît. Choisissez la dernière version du container via "**latest**" et validez.

image.png

Quand le téléchargement est effectué, cliquez à gauche sur "**Conteneur**" dans l'interface de Container Manager. Créez un nouveau conteneur en cliquant sur "**Créer**".

Commencez par choisir l'image "**vaultwarden/server:latest**" que vous venez de télécharger. Même si ce n'est pas obligatoire, vous pouvez **activer la limitation des ressources** comme sur l'image ci-dessous. Ce conteneur n'est pas très gourmand. Au besoin, c'est ajustable par la suite. Activez aussi le redémarrage automatique pour que le NAS essaie de le relancer automatiquement en cas de crash.

image.png

L'étape "**Paramètres des ports**" s'affiche. Les ports "**3012/TCP**" et "**80/TCP**" correspondent aux deux ports utilisés par le conteneur. Celui qui correspond à l'accès à l'interface Web, c'est bien sûr le port 80 correspondant au HTTP. L'autre port correspond au Websocket. Tout à gauche, c'est le port à attribuer au niveau du NAS.

Dans l'exemple ci-dessous, l'accès au NAS sur **le port 3012/TCP va renvoyer vers le port 3012/TCP** du conteneur. Tandis que **le port 3013/TCP va renvoyer vers le port 80/TCP** du conteneur. Vous devez utiliser des ports qui ne sont pas encore utilisés sur votre NAS (il y a des chances pour que le port 80 soit utilisé par un autre service). Vous pouvez personnaliser ces valeurs.

image.png

Vous devez prévoir un espace de stockage sur votre NAS afin que le container Vaultwarden puisse écrire ses données. Dans le répertoire "**docker**" du NAS, créez un répertoire nommé "**vaultwarden**" et vous allez ensuite l'associer au container.

image.png

Dans la configuration du container, cliquez sur "ajouter un dossier", sélectionnez le répertoire "**/docker/vaultwarden**" et montez-le en "**/data**" afin que le container stocke ses données à cet endroit. Il est important de sauvegarder le répertoire "**/docker/vaultwarden**" pour protéger votre coffre-fort (vous pouvez utiliser Hyper Backup, par exemple).

Ne touchez pas à la section "**Environnement**", nous allons revenir dessus par la suite.

image.png

Il n'est pas nécessaire de configurer de fonctionnalités ou d'adapter la configuration réseau. Cliquez sur "**Suivant**".

image.png

Cliquez sur "**Effectué**" pour créer le conteneur Vaultwarden !

image.png

La section "**Conteneur**" contient un petit nouveau nommé "**vaultwarden**" et qui est actif.

image.png

Afin de pouvoir accéder à l'interface Web de Vaultwarden, vous devez **autoriser le port 3013/TCP** (ou autre, selon votre configuration) dans le **pare-feu de DSM**. Enfin, ceci est vrai uniquement si vous avez activé et configuré le pare-feu de DSM, ce qui est recommandé. Pour rappel, on accède au pare-feu de cette façon : **Panneau de configuration > Sécurité > Pare-feu**.

image.png

Une fois la règle de pare-feu créée, il est possible d'accéder à Vaultwarden via un navigateur en précisant l'adresse IP du NAS et le port 3013.

image.png