

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

Révision #1

Créé 11 janvier 2025 19:51:32 par Angelo

Mis à jour 1 avril 2025 18:42:33 par Angelo