

docker

SOMMAIRE

Qu'est ce que docker	3
Installation de Docker	3
Installation d'un Conteneur	3
Vérification du fonctionnement	5
Qu'est ce que Portainer	5
Les avantages de Portainer	5
Installation de Portainer	6
Utilisation de Portainer	6
Qu'est ce qu'un Stack	8
Création d'un Stack	8
Déploiement de Nextcloud	8
Qu'est ce que Nextcloud	8
Analyse du fichier docker-compose.yml de Nextcloud	9
Vérification du fonctionnement	9
Déploiement de Nginx et MariaDB	10
Qu'est ce que MariaDB	10
Vérification du fonctionnement de NGINX	11
Vérification du fonctionnement de MariaDB	11
Création d'un Network pour 2 stacks	12

Qu'est ce que docker

Docker est une plateforme open-source qui permet de développer, déployer et exécuter des applications dans des conteneurs. Un **conteneur Docker** encapsule une application avec tous ses dépendances logicielles, ce qui permet à l'application de s'exécuter de manière cohérente et isolée sur n'importe quel environnement compatible avec **Docker**, qu'il s'agisse d'un ordinateur portable, d'un serveur physique ou d'un cloud public.

En utilisant **Docker**, les développeurs peuvent packager leurs applications avec toutes leurs dépendances dans des conteneurs légers et portables. Cela facilite le déploiement et la gestion des applications, car les conteneurs **Docker** garantissent la cohérence entre les environnements de développement, de test et de production.

Installation de Docker

Mise à jour des paquets : **apt update**

Installation des paquets nécessaires : **wget <https://get.docker.com/>**

Ensuite, taper cette commande pour terminer l'installation

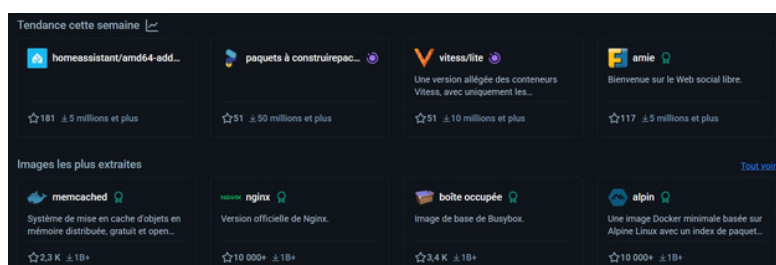
```
root@bullseye:~# bash index.html
```

Ensuite, nous pouvons vérifier l'état du service Docker:

```
root@bullseye:~# systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2026-04-09 20:07:54 CEST; 23s ago
  TriggeredBy: • docker.socket
  Docs: https://docs.docker.com
  Main PID: 7763 (dockerd)
  Tasks: 8
  Memory: 26.8M
  CPU: 239ms
  CGroup: /system.slice/docker.service
          └─7763 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Installation d'un Conteneur

Maintenant que Docker est installé, nous pouvons installer des conteneurs à partir d'une image. Pour télécharger une image, nous nous rendons sur **Docker Hub**.



Nous pouvons voir qu'il y'a des milliers d'images, nous allons nous intéresser à une en particulier. Si nous prenons l'image de DokuWiki, un site web collaboratif, il suffira juste de prendre le lien de l'image et de la rajouter à la commande **docker pull**

```
Commande Docker Pull
```

```
docker pull dokuwiki/dokuwiki
```

Copie

```
root@buster:~# docker pull mprasil/dokuwiki
Using default tag: latest
latest: Pulling from mprasil/dokuwiki
2b55860d4c66: Pull complete
2d785b3b2b4c: Pull complete
e8c2ce754318: Pull complete
a016a328fe36: Pull complete
74d3af8881ec: Pull complete
77e015da3e46: Pull complete
1b00e5cae740: Pull complete
7c7e2977ff067: Pull complete
```

Une fois l'image téléchargée et installée théoriquement, vous pouvez vérifier sa présence dans le dépôt d'images locales à l'aide de la commande '**docker images**'. Cette commande vous affichera la liste de toutes les images **Docker** installées sur votre système, y compris l'image '**mprasil/dokuwiki**' si elle a été téléchargée avec succès.

```
root@buster:~# docker images
```

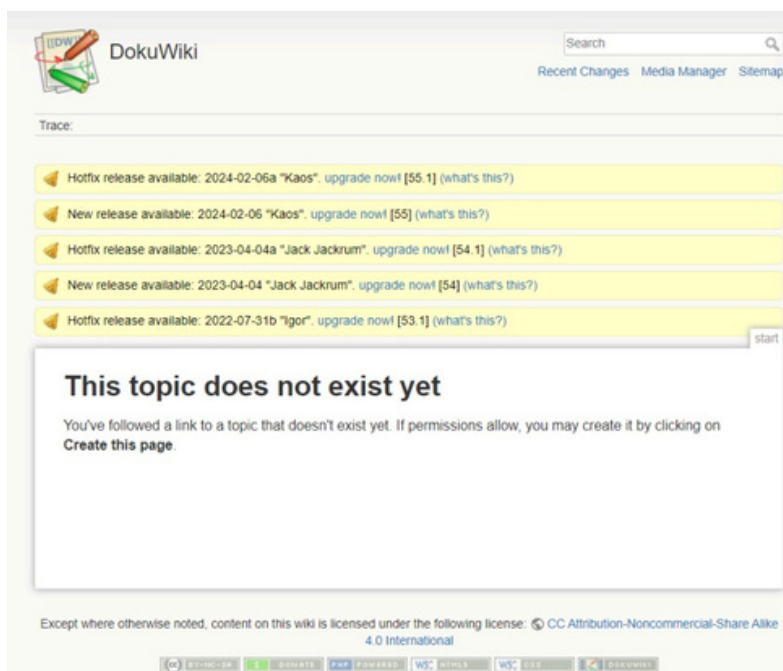
*REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mprasil/dokuwiki	latest	78c138e0b03c	16 months ago	319MB

```
root@buster:~# docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki
7556305e00c0fcd1af1f931164b199edbe26cfd41b998964c640f3fe6b958e31
```

- '-d' : Indique un lancement en arrière plan
- '-p 8001:80' : redirige le port 8001 de la machine physique vers le port 80 du conteneur
- '--name wiki01' : Affecte un nom au conteneur. En l'occurrence Wiki01
- 'mprasil/dokuwiki' : Nom de l'image sur le hub docker

Vérification du fonctionnement

Dans le navigateur, tapez “**192.168.56.55:8081**”



Qu'est ce que Portainer

Portainer est une interface graphique qui facilite la gestion des conteneurs Docker. À la base, Docker fonctionne principalement en ligne de commande, ce qui peut être complexe à gérer, surtout avec plusieurs conteneurs, volumes et réseaux.

Les avantages de Portainer

Interface graphique intuitive : Gère Docker sans ligne de commande, tout est accessible via le web.

Déploiement rapide : Lance des conteneurs et stacks en quelques clics, compatible Docker Swarm et Kubernetes.

Monitoring intégré : Suivi des conteneurs, logs et utilisation des ressources en temps réel.

Installation de Portainer

Tutoriel d'installation : <https://docs.portainer.io/start/install-ce/server/docker/linux>

Création d'un volume pour portainer

```
docker volume create portainer_data
```

Lancer le conteneur Portainer

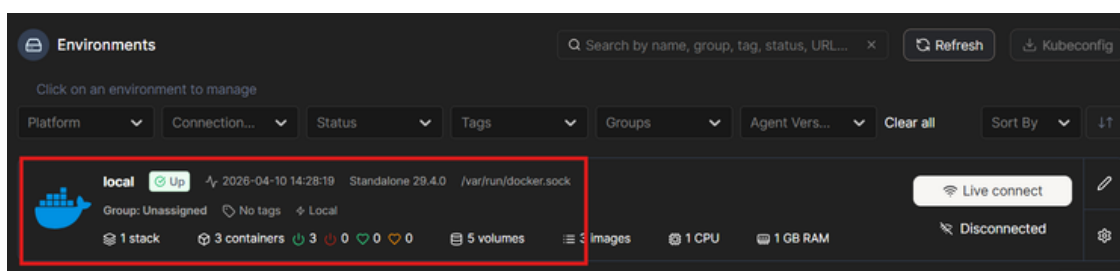
```
docker run -d \  
  --name portainer \  
  --restart=always \  
  -p 8000:8000 \  
  -p 9443:9443 \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  -v portainer_data:/data \  
  portainer/portainer-ce:latest
```

Accès à l'interface Web

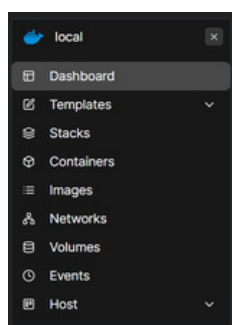
```
https://<IP_DU_SERVEUR>:9443
```

Utilisation de Portainer

Quand on se connecte en Admin, il faut se rendre dans l'onglet **local**

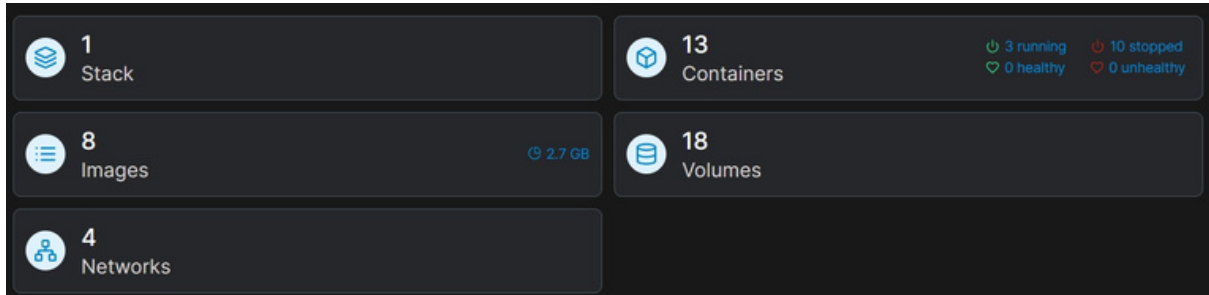


Dans l'onglet local, nous avons accès à beaucoup de fonctionnalités



Voici ce que les fonctionnalités dans 'Local' permettent de faire

Dashboard : Vue d'ensemble des ressources Docker, des conteneurs actifs et des performances.



Templates : Modèles prédéfinis pour déployer rapidement des applications.

Stacks : Gestion et déploiement de services avec Docker Compose ou Swarm.

Containers : Supervision et contrôle des conteneurs (démarrage, arrêt, logs, terminal).

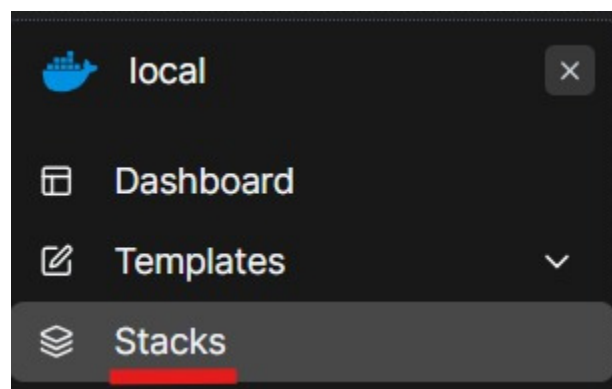
Images : Téléchargement, stockage et gestion des images Docker.

Networks : Configuration et administration des réseaux Docker.

Volumes : Gestion du stockage persistant utilisé par les conteneurs.

Events : Suivi des actions et événements liés aux conteneurs et services.

Nous allons nous rendre dans l'onglet **Stacks**



Qu'est ce qu'un Stack

Une **Stack** est un ensemble de services Docker déployés ensemble à l'aide d'un fichier **Docker Compose**. Elle permet de gérer plusieurs conteneurs qui interagissent entre eux (base de données, backend, frontend, etc.) de manière simplifiée.

Création d'un Stack

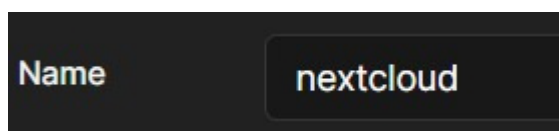
Nous allons créer un nouveau Stack, il faut se rendre sur 'add stacks'

Déploiement de Nextcloud

Qu'est ce que Nextcloud

Nextcloud est une solution open-source d'hébergement de fichiers et de collaboration, offrant une alternative sécurisée à Google Drive ou Dropbox. Il permet destocker, partager et synchroniser ses données tout en gardant le contrôle sur la confidentialité.

Il faut donner un nom au stacks, dans notre cas, nous allons l'appeler '**nextcloud**'



En créant le **stack**, nous allons importer son fichier Docker Compose:

```
version: '3.0'
services:
  db:
    image: mariadb:latest
    restart: always
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    environment:
      MYSQL_ROOT_PASSWORD: rootpass
      MYSQL_DATABASE: nextcloud
      MYSQL_USER: nextcloud
      MYSQL_PASSWORD: password
    volumes:
      - db_data:/var/lib/mysql

  nextcloud:
    image: nextcloud:latest
    restart: always
    ports:
      - "8080:80"
    environment:
      NEXTCLOUD_ADMIN_USER: admin
      NEXTCLOUD_ADMIN_PASSWORD: adminpass
      MYSQL_HOST: db
      MYSQL_DATABASE: nextcloud
      MYSQL_USER: nextcloud
      MYSQL_PASSWORD: password
    volumes:
      - nextcloud_data:/var/www/html

volumes:
  db_data:
  nextcloud_data:
```

Analyse du fichier docker-compose.yml de Nextcloud

Dans ce fichier, nous retrouvons la version **3.9**, la **base de données** utilisant l'image **MariaDB**, l'**image Nextcloud** qui constitue la partie principale, ainsi que le **port 8080:80** pour l'accès à l'application.

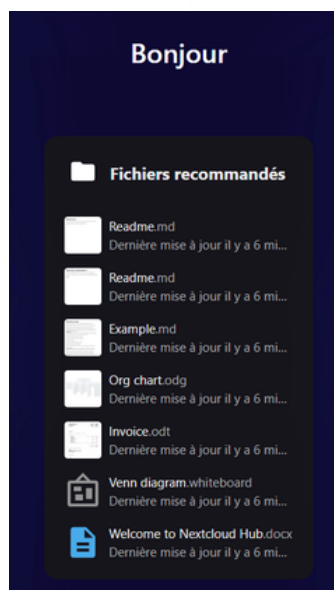
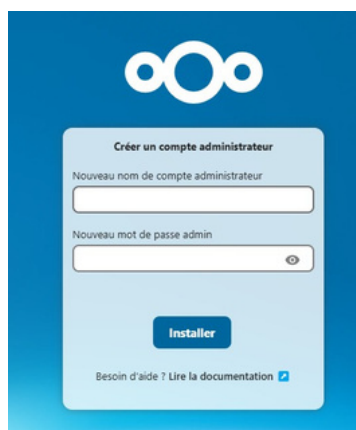
Vérification du fonctionnement

Une fois l'installation terminée, nous pouvons voir dans la catégorie 'stack ' que **Nextcloud** est bien présent

Pour accéder à **Nextcloud**, il suffit d'entrer son adresse IP, par exemple

192.168.56.155, suivie du port **8080**, soit : <http://192.168.56.155:8080>

Nous accédons à **Nextcloud**



Déploiement de Nginx et MariaDB

Qu'est ce que MariaDB

MariaDB est un système de gestion de base de données édité sous licence GPL. Il s'agit d'un dérivé communautaire de MySQL : la gouvernance du projet est assurée par la fondation MariaDB, et sa maintenance par la société Monty Program AB.

Il faut donner un nom au stack, dans notre cas, nous allons l'appeler **nginx-mariadb**

The logo consists of the text 'nginx-mariadb' in a light blue, monospace-style font, centered on a dark grey rectangular background.

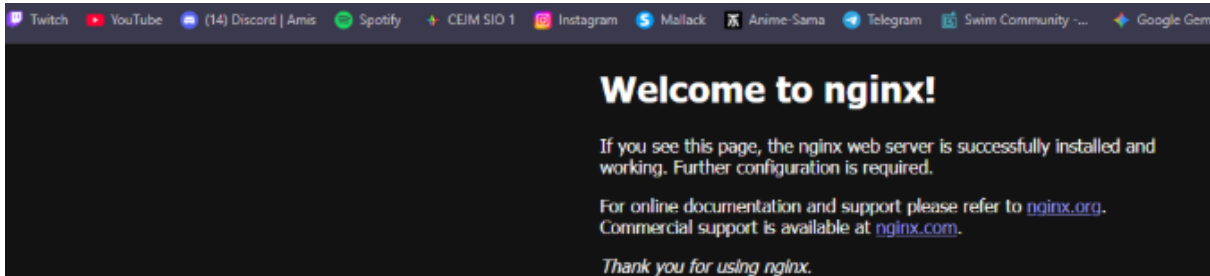
Ensuite, nous venons ajouter notre **fichier compose**, ce fichier est basé sur **Docker Compose** et contient des instructions sur les conteneurs à déployer, leurs configurations, leurs dépendances et les ports exposés.

```
1 version: '3.8'
2
3 services:
4   # Votre serveur Web
5   web:
6     image: nginx:1.25-alpine
7     container_name: nginx-web
8     ports:
9       - "8085:80"
10    volumes:
11      - /opt/nginx-content/html:/usr/share/nginx/html:ro
12    networks:
13      - app-network
14    restart: always
15
16   # Votre Base de Données
17   db:
18     image: mariadb:10.11
19     container_name: mariadb-db
20     restart: always
21     environment:
22       MARIADB_ROOT_PASSWORD: mot_de_passe_robuste
23       MARIADB_DATABASE: mon_site
24       MARIADB_USER: utilisateur
25       MARIADB_PASSWORD: mot_de_passe_utilisateur
26     volumes:
27       - db_data:/var/lib/mysql
28     networks:
29       - app-network
30
31 networks:
32   app-network:
33     driver: bridge
34
35 volumes:
36   db_data:
```

```
adminer:
  image: adminer
  restart: always
  ports:
    - 8086:8080
  networks:
    - app-network
```

Vérification du fonctionnement de nginx

Une fois l'installation terminée, nous pouvons vérifier que le conteneur Nginx-MariaDB fonctionne correctement et qu'il est bien accessible via son port dédié.



Vérification du fonctionnement de mariadb

Une fois l'installation terminée, nous pouvons vérifier que le conteneur Nginx-MariaDB fonctionne correctement et qu'il est bien accessible via son port dédié.

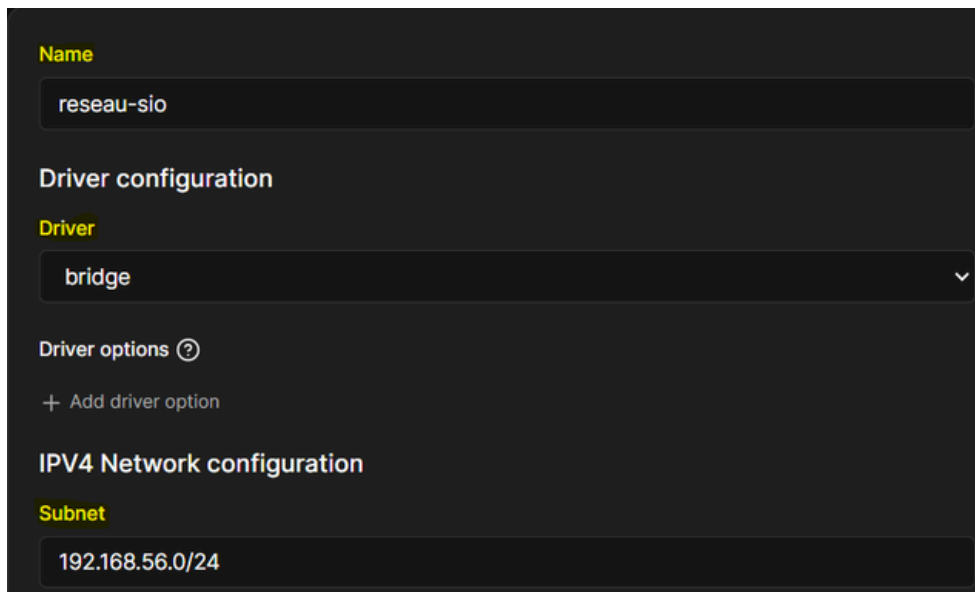


Création d'un Network pour 2 stacks

Se rendre dans Network → Add Network

+ Add network

Ensuite, nous allons configurer le réseau de cet manière



The screenshot shows a dark-themed configuration form for a network. It includes the following sections and fields:

- Name:** A text input field containing the value "reseau-sio".
- Driver configuration:**
 - Driver:** A dropdown menu with "bridge" selected.
 - Driver options:** A section with a question mark icon and a "+ Add driver option" link.
- IPV4 Network configuration:**
 - Subnet:** A text input field containing the value "192.168.56.0/24".