

# Mise en service de TTRSS avec Docker



Dans cette documentation nous installerons le logiciel TTRSS avec Docker.

Pour mieux s'y retrouver cette documentation disposera de plusieurs screenshots illustrant les consignes.

## Préambule

### Présentation :

Tiny Tiny RSS (souvent abrégé en TTRSS) est un lecteur et agrégateur RSS/ATOM

Accessible ensuite depuis n'importe quel navigateur, il permet de s'abonner aux flux RSS ou ATOM des sites que vous souhaitez suivre afin de recevoir les derniers articles publiés. Une application officielle android est également disponible.

Il propose de nombreuses fonctionnalités telles que l'import / export OPML, le partage via RSS ou réseaux sociaux, l'ajout de plugins ou de thèmes, le support des podcasts, le filtrage des flux RSS, etc.

Nous considérons que vous êtes équipé de cette manière :

1. Une VM sous Debian avec docker à installer **[node03]**

Les allocations de matériel (CPU/RAM...) sont à allouer selon vos envies, attention à respecter la configuration minimale. C'est à dire :

Pour **TTRSS** :

1. *512MB* de ram

2. 1 cœurs de CPU
3. 8GB d'espace disque
4. *Debian 11*
5. *PHP 7.4*
6. *Docker*

Nos IP pour notre infrastructure seront :

1. [TTRSS] : **10.192.100.123**

Mot de passe par défaut sur toutes les sessions : **Not24get**

Rappel des deux commandes essentiels :

1. `ip a` (connaitre son adresse IP)
2. `nano /etc/network/interfaces` (configuration de l'interface réseau)

**Conseil** : Ajouter les deux machines dans un logiciel tel que mRemoteNG pour faciliter l'administration.

# Installation de TTRSS

## Installation de Docker via le repo

### Ajout du dépôt

- Ajouter les paquets pour installer des paquets depuis des sources sécurisées

```
apt-get install \
    ca-certificates \
    curl \
    gnupg
```

- Ajouter la clé GPG de Docker

```
mkdir -m 0755 -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
```

- Ajouter le repo dans le `sources.list`

```
echo \
    "deb [arch=$(dpkg --print-architecture)] signed-
    by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
```

```
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
tee /etc/apt/sources.list.d/docker.list > /dev/null
```

## Installation du moteur Docker

- Mettre à jour le nouveau dépôt

```
apt-get update
```

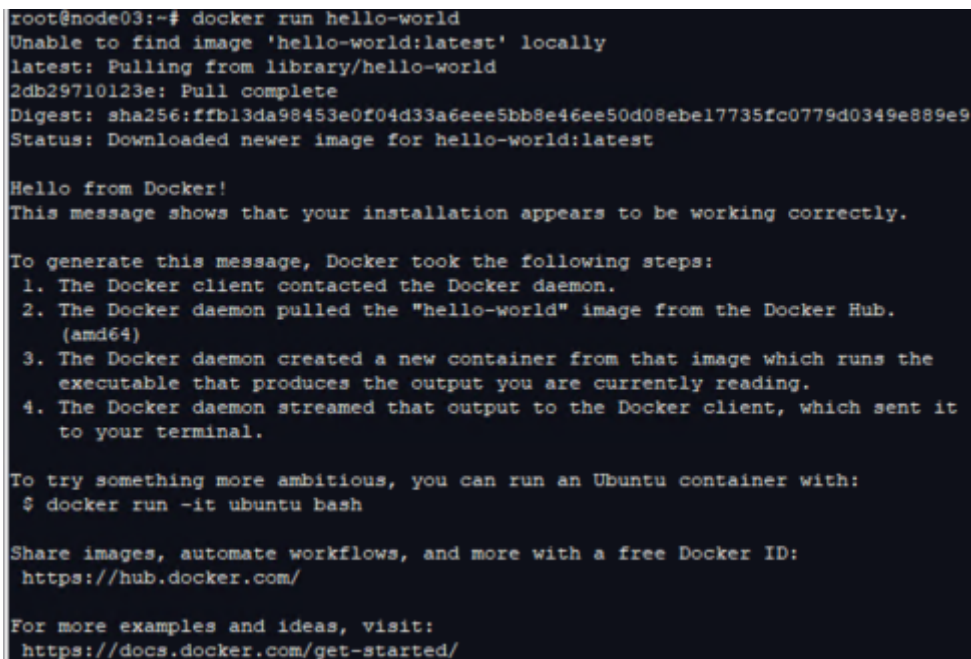
- Installation de Docker et de ses dépendances

```
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

## Essais

Cette commande télécharge une image de test et l'exécute dans un conteneur. Lorsque le conteneur s'exécute, il imprime un message de confirmation et se ferme.

```
docker run hello-world
```

A terminal window showing the output of the 'docker run hello-world' command. The output indicates that the 'hello-world:latest' image was pulled from the Docker Hub and a new container was created and executed. The container prints a message: 'Hello from Docker! This message shows that your installation appears to be working correctly.' It then lists the steps Docker took to generate this message: 1. The Docker client contacted the Docker daemon. 2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64) 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading. 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal. It also provides instructions on how to try something more ambitious by running an Ubuntu container with 'docker run -it ubuntu bash'. Finally, it shares a Docker ID and links to the Docker Hub and documentation.

```
root@node03:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:fffb13da98453e0f04d33a6eee5bb8e46ee50d08eb17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

## Installation de TTRSS

### Depuis le repo Github

Cette image embarque un serveur web sous base nginx et un serveur SQL.

- ```
git clone https://git.tt-rss.org/fox/ttrss-docker-compose.git ttrss-docker
```

```
cd ttrss-docker
```

- Changer de branche vers `static-dockerhub`

```
git checkout static-dockerhub
```

- Configuration de l'environnement

```
mv .env-dist .env
nano .env
```

IMAGE

Modifier le fichier selon votre volonté

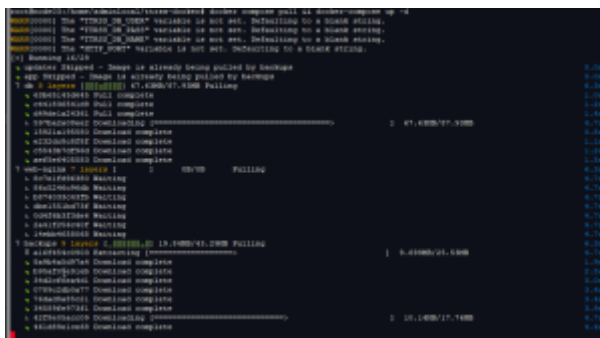


Attention à bien spécifier l'adresse EXACTE à partir de laquelle ttrss sera accessible (TTRSS\_SELF\_URL\_PATH=) Définissez un mot de passe admin à la ligne (ADMIN USER PASS=)

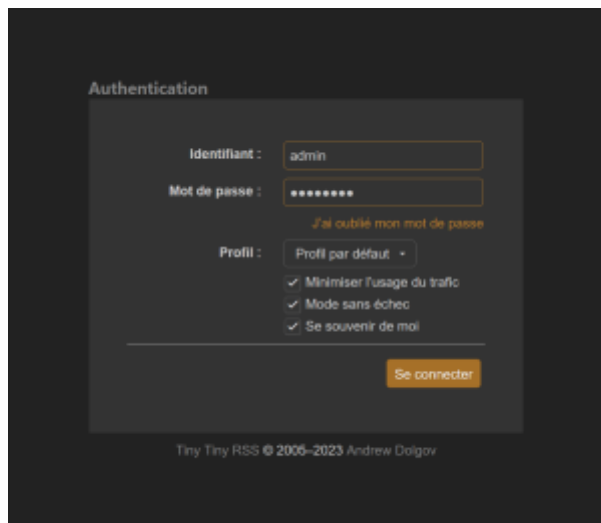
```
# Set admin user password to this value. If not set, random password will be
# generated if default password is being used, look for it in the 'app'
# container logs.
ADMIN_USER_PASS=password
```

- Télécharger et démarrer le container

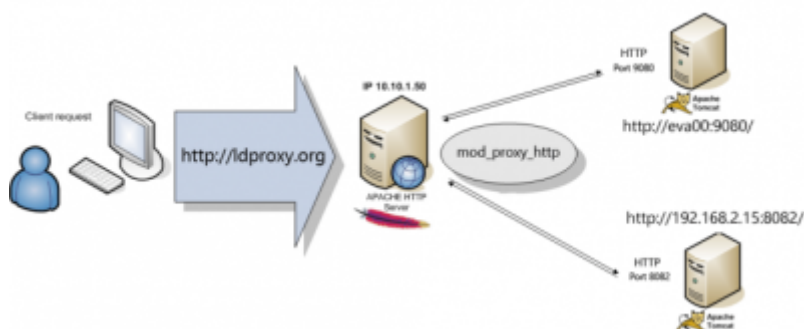
```
docker compose pull && docker compose up -d
```



Visiter la page : <http://<IP>:8280/tt-rss>



## Reverse proxy avec nginx



A partir d'un autre serveur nginx.

## Création de la config nginx

- Créer une entrée A dans votre serveur DNS qui point vers l'ip de votre serveur
  - Exemple :

|                    |    |   |                |
|--------------------|----|---|----------------|
| ttrss.stoneset.fr. | IN | A | 10.192.100.209 |
|--------------------|----|---|----------------|

- Créer la configuration dans les *sites-available* :

```
nano /etc/nginx/sites-available/proxy_ttrss.conf
```

[snippet.sh](#)

```
server {
    server_name ttrss.stoneset.fr;
    location / {
        proxy_pass http://node03.local:8280;
        proxy_http_version 1.1;
    }
}
```

```
proxy_set_header Upgrade $http_upgrade;  
proxy_set_header Connection 'upgrade';  
proxy_set_header Host $host;  
proxy_cache_bypass $http_upgrade;  
}  
}
```



Spécifier un chemin relatif pour le lien proxy\_pass, il est défini dans /etc/hosts

- Redémarrer nginx :

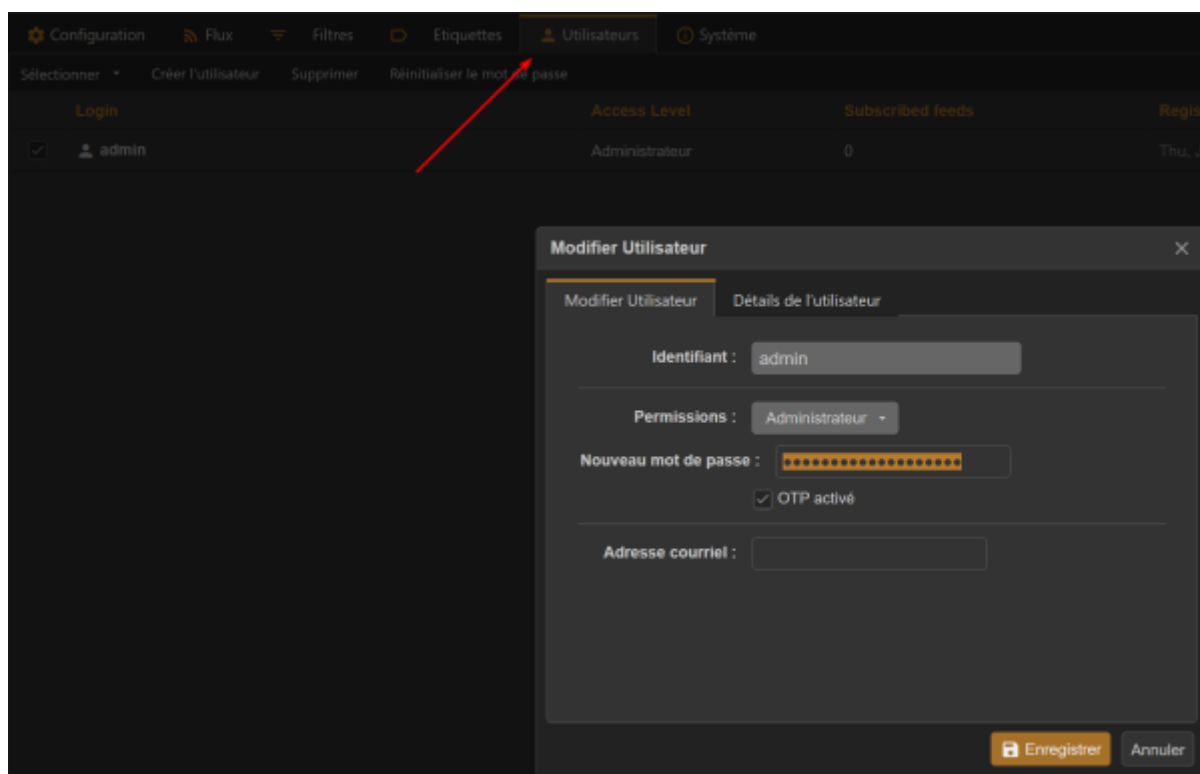
```
systemctl restart nginx
```

## Configuration de TTRSS:

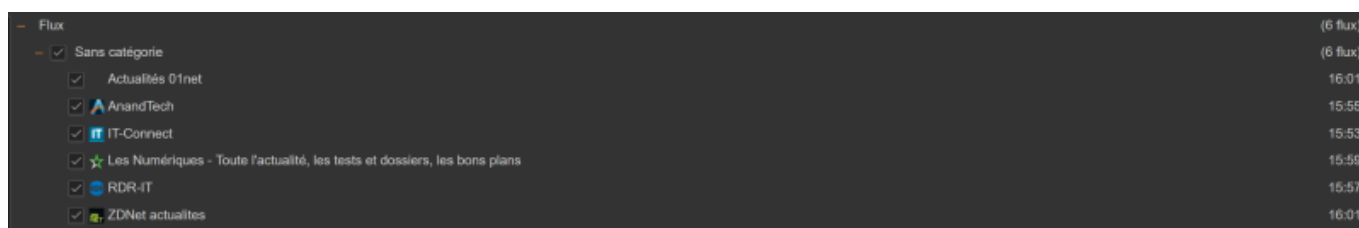
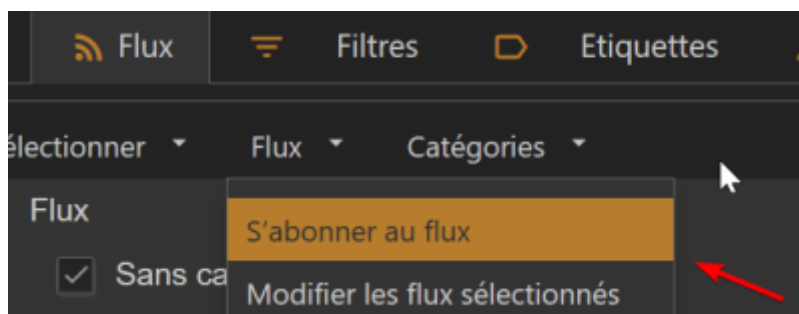
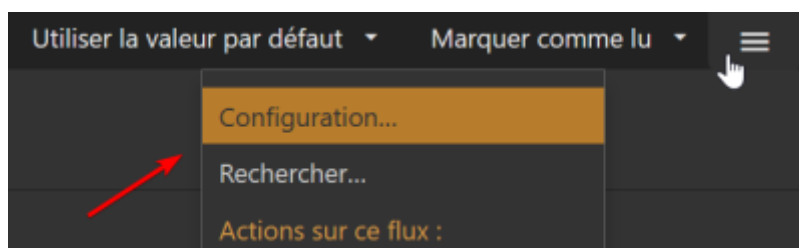


Le compte par défaut est admin:password

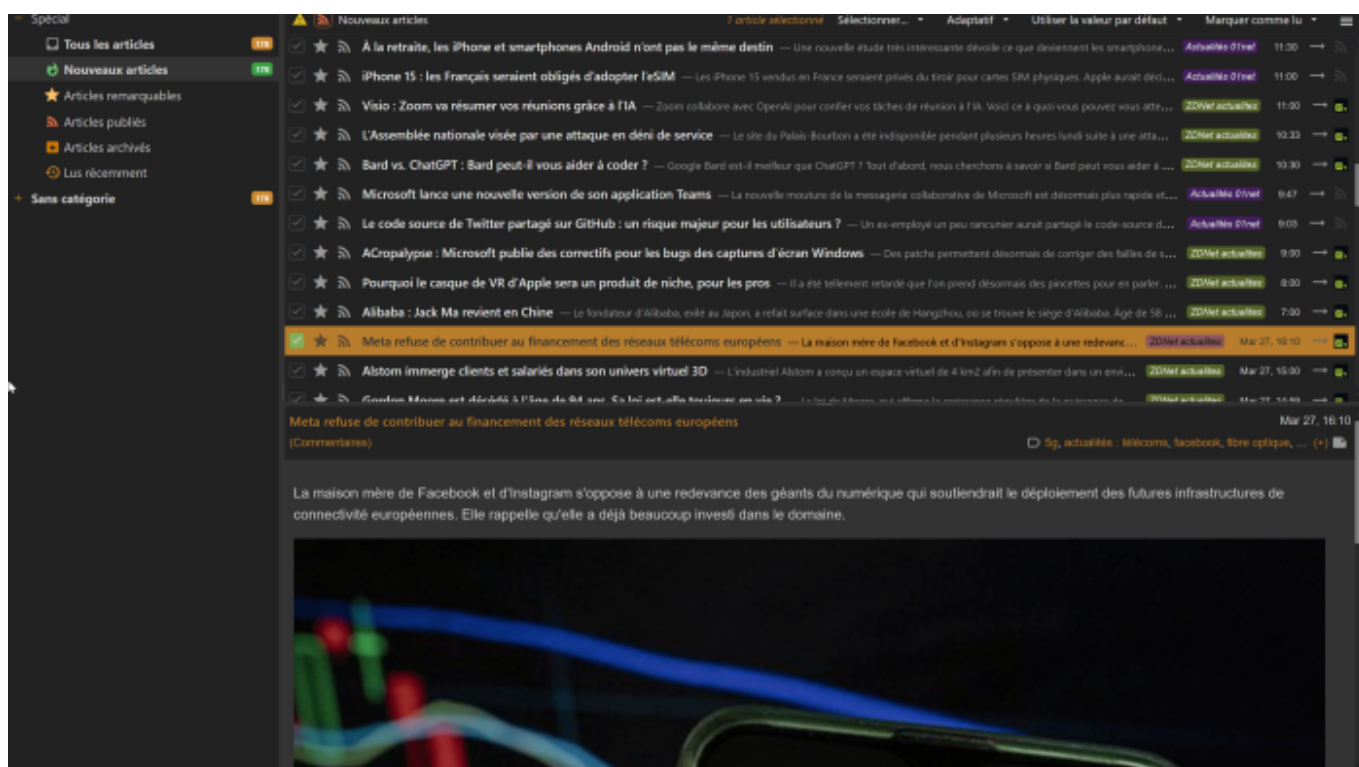
- Changer le mot de passe par défaut



## Ajouter un flux RSS



Une fois les flux ajoutés :



- Liste de flux well-known RSS :
  - <https://www.anandtech.com/rss>
  - <https://www.it-connect.fr/feed/>
  - <https://rdr-it.com/feed/>
  - <https://www.lesnumeriques.com/rss.xml>
  - <https://www.zdnet.fr/feeds/rss/actualites/>
  - <https://www.01net.com/actualites/feed/>
  - <https://www.lemondeinformatique.fr/flux-rss/thematique/telecom/rss.xml>

## Mes sources

1. <https://framalibre.org/content/tiny-tiny-rss>
2. <https://tt-rss.org/wiki/InstallationNotes>
3. <https://docs.docker.com/engine/install/debian/>

From:  
<https://wiki.stoneset.fr/> - **StoneSet - Documentations**

Permanent link:  
[https://wiki.stoneset.fr/doku.php?id=wiki:linux:ttrss\\_installation&rev=1680013064](https://wiki.stoneset.fr/doku.php?id=wiki:linux:ttrss_installation&rev=1680013064)

Last update: **2023/03/28 14:17**

