

FTP

Le protocole FTP (File Transfer Protocol) est un protocole utilisé pour le transfert de fichiers entre un client et un serveur sur un réseau. Il nous permet d'envoyer, de récupérer et de gérer des fichiers sur un serveur distant que je vais devoir créer sous forme **containerisée**.

Mise en service

On a utilisé une VM FEDORA. Je vais y installer docker et tous les paquets nécessaires

```
sudo dnf -y install dnf-plugins-core
sudo dnf config-manager --add-repo
https://download.docker.com/linux/fedora/docker-ce.repo
sudo dnf -y install docker-ce docker-ce-cli containerd.io
sudo systemctl enable --now docker
```

```
[iut@localhost ~]$ sudo dnf update
[sudo] Mot de passe de iut :
Dernière vérification de l'expiration des métadonnées effectuée le mer. 07 juin 2023 08:22:26.
Dépendances résolues.
Rien à faire.
Terminé !
```

```
[root@localhost iut]# sudo dnf -y install dnf-plugins-core
sudo dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
sudo dnf install docker-ce docker-ce-cli containerd.io
Dernière vérification de l'expiration des métadonnées effectuée il y a 3:28:44 le mer. 07 juin 2023 08:22:26.
Le paquet dnf-plugins-core-4.4.1-1.fc38.noarch est déjà installé.
Dépendances résolues.
Rien à faire.
Terminé !
Ajout du dépôt depuis : https://download.docker.com/linux/fedora/docker-ce.repo
Docker CE Stable - x86_64 49 kB/s | 3.5 kB 00:00
Le paquet docker-ce-3:24.0.2-1.fc38.x86_64 est déjà installé.
Le paquet docker-ce-cli-1:24.0.2-1.fc38.x86_64 est déjà installé.
Le paquet containerd.io-1.6.21-3.1.fc38.x86_64 est déjà installé.
Dépendances résolues.
Rien à faire.
Terminé !
[root@localhost iut]#
```

```
[root@localhost iut]# sudo systemctl start docker
sudo systemctl enable docker
[root@localhost iut]#
```

- J'installe aussi docker compose

```
[root@localhost FTP]# sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
0	0	0	0	0	--:--:--	0:00:05	0
0	0	0	0	0	--:--:--	0:00:05	0
100 52.0M	100 52.0M	0	4989k	0	0:00:10	0:00:10	10.3M

```
[root@localhost FTP]#
```

```
[root@localhost FTP]# sudo chmod +x /usr/local/bin/docker-compose
[root@localhost FTP]# docker-compose --version
Docker Compose version v2.18.1
[root@localhost FTP]#
```

Fichier de configuration

Voici le “[docker-compose.yaml](#)” qui va mettre en place un serveur FTP

```
version: '3'

services:
  ftpd_server:
    image: stillliard/pure-ftpd
    container_name: pure-ftpd
    ports:
      - "21:21"
      - "30000-30009:30000-30009"
    mac_address: 02:42:c0:a8:84:47
    networks:
      x_vlan:
        ipv4_address: 192.168.16.47
    volumes:
      - ./data:/home/ftpusers
    environment:
      FTP_USER_NAME: ftpuser
      FTP_USER_PASS: ftproot
      FTP_USER_HOME: /home/ftpusers
    restart: always

networks:
  x_vlan:
    driver: macvlan
    driver_opts:
      parent: ens18
    ipam:
```

```
driver: default
config:
  - subnet: 192.168.16.0/24
    gateway: 192.168.16.99
```

Ce fichier Docker Compose définit un service pour exécuter un serveur FTP en utilisant l'image Docker [stilliard/pure-ftpd](#). Voici une explication ligne par ligne du contenu du fichier :

- On utilise la version 3 de [Compose](#)
- On crée un service appelé [ftpd_server](#) utilisant l'image [stilliard/pure-ftpd](#).
- [pure-ftpd](#) signifie le nom du container

```
[iut@localhost ftp]$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS        NAMES
32aaf11e550d   stilliard/pure-ftpd "/bin/sh -c '/run.sh..." 25 hours ago   Up 25 hours   pure-ftpd
```

- le conteneur écoutera les connexions entrantes sur ces ports: [21](#) et [30000-30009](#)

```
ports:
  - "21:21"
  - "30000-30009:30000-30009"
```

- On définit l'adresse MAC du conteneur sur [02:42:c0:a8:84:47](#).
- On connecte le service au réseau [x_vlan](#).
- On attribue au conteneur une adresse IP spécifique dans le réseau [x_vlan](#): [192.168.16.47](#).

```
mac_address: 02:42:c0:a8:84:47
networks:
  x_vlan:
    ipv4_address: 192.168.16.47
```

- Pour persister les données du serveur FTP, on monte un volume. Le dossier local [./data](#) est monté dans le répertoire [/home/ftpusers](#) du conteneur.

```
volumes:
  - ./data:/home/ftpusers
```

- On définit des variables d'environnement pour configurer le nom d'utilisateur, le mot de passe et le répertoire de base du serveur FTP.

```
environment:
  FTP_USER_NAME: ftpuser
  FTP_USER_PASS: ftproot
  FTP_USER_HOME: /home/ftpusers
```

- On configure le service de façon à redémarrer automatiquement en cas d'échec.

```
restart: always
```

- Enfin, on définit une configuration pour le réseau `x_vlan`. Il utilise le pilote `macvlan` et spécifie `ens18` comme interface parente. De plus, on va définir une plage d'adresses IP pour le réseau, avec une adresse de sous-réseau `192.168.16.0/24` et une passerelle `192.168.16.99`.

```
networks:
  x_vlan:
    driver: macvlan
    driver_opts:
      parent: ens18
    ipam:
      driver: default
      config:
        - subnet: 192.168.16.0/24
          gateway: 192.168.16.99
```

En résumé, ce fichier Docker Compose crée un conteneur de serveur FTP utilisant l'image `stilliard/pure-ftpd`, expose les ports nécessaires, configure un réseau spécifique et monte un volume pour les données du serveur FTP.

Je vais maintenant lancer mon docker-compose: `docker-compose up -d`

```
[root@localhost FTP]# sudo docker-compose up -d
[+] Running 11/11
✓ ftp 10 layers [██████████] 0B/0B Pulled 11.5s
✓ b4d181a07f80 Pull complete 7.3s
✓ e98cb485cfd8 Pull complete 8.3s
✓ 0822c77e0e0b Pull complete 8.3s
✓ 8588bb5b4480 Pull complete 8.4s
✓ a41cb6218cc9 Pull complete 10.0s
✓ 7ac94abef10b Pull complete 10.0s
✓ 8d5dc14fad00 Pull complete 10.1s
✓ 417d6be4b5ec Pull complete 10.1s
✓ 71cd100c9ec0 Pull complete 10.1s
✓ 227193214c34 Pull complete 10.1s
[+] Building 0.0s (0/0)
[+] Running 2/2
✓ Network ftp_default Created 0.1s
✓ Container ftp-ftp-1 Started 1.6s
```

Vérification

Voici comment fonctionne généralement une connexion FTP :

1. Établissement de la connexion : je me connecte(client FTP) au serveur FTP en utilisant l'adresse IP du serveur.

2. Authentification : je fournis les informations d'identification (nom d'utilisateur et mot de passe) pour s'authentifier auprès du serveur FTP. Ces informations sont configurées dans l'environnement du serveur FTP (comme dans le fichier Docker Compose mentionné précédemment).

3. Navigation dans les répertoires : Une fois authentifié, je peux naviguer dans les répertoires du serveur FTP à l'aide de commandes FTP telles que `cd` (changement de répertoire), `ls` (liste des fichiers et répertoires), `pwd` (affiche le répertoire de travail actuel), etc.

4. Transfert de fichiers : je peux envoyer des fichiers au serveur FTP en utilisant la commande `put` ou récupérer des fichiers du serveur FTP en utilisant la commande `get`. Les fichiers sont transférés entre le client et le serveur via la connexion FTP établie.

Pour vérifier le bon fonctionnement de mon serveur FTP, j'ouvre un terminal et j'essaye d'y accéder

```
test@232-22:~$ ftp 192.168.16.47
Connected to 192.168.16.47.
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 5 allowed.
220-Local time is now 10:12. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (192.168.16.47:test): ftpuser
331 User ftpuser OK. Password required
Password:
230 OK. Current directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
```

Excellent!!!! connexion réussie.

Maintenant je n'ai qu'à essayer de faire passer quelque commande pour bien vérifier

```

ftp> ls
200 PORT command successful
150 Connecting to port 55237
drwxr-xr-x  2 1000      ftpgroup      6 Jun 13 08:49 essai
drwxr-xr-x  2 1000      ftpgroup      6 Jun 13 08:49 test
226-Options: -l
226 2 matches total
ftp> mkdir ftp_essai
257 "ftp_essai" : The directory was successfully created
ftp> ls -al
200 PORT command successful
150 Connecting to port 40801
drwxr-xr-x  5 1000      0              48 Jun 14 10:13 .
drwxr-xr-x  5 1000      0              48 Jun 14 10:13 ..
drwxr-xr-x  2 1000      ftpgroup      6 Jun 13 08:49 essai
drwxr-xr-x  2 1000      ftpgroup      6 Jun 14 10:13 ftp_essai
drwxr-xr-x  2 1000      ftpgroup      6 Jun 13 08:49 test
226-Options: -a -l
226 5 matches total
ftp> pwd
257 "/" is your current location
ftp>

```

J'ai fait un mkdir de ftp_essai et je vois que j'arrive à créer un répertoire, je pourrais même éventuellement le déplacer. Le commande pwd marche bien aussi.

```

ftp> pwd
257 "/" is your current location
ftp> cd ftp_essai
250 OK. Current directory is /ftp_essai
ftp> pwd
257 "/ftp_essai" is your current location
ftp>

```