

SERVEUR FTP

Pour faire le serveur FTP, j'ai décidé de le dockeriser en créant un fichier qui se nomme Dockerfile et en l'utilisant ensuite pour construire notre image pour le container.

Dans un Dockerfile, on a plusieurs blocs qui reviennent fréquemment. On a une première ligne FROM qui permet de définir l'image de base à utiliser par le container. Ensuite nous avons RUN qui permet de définir les commandes à exécuter dans le container. Puis nous avons EXPOSE qui permet de préciser sur quel port le container va écouter. Et enfin CMD qui permet de donner la commande que doit exécuter le container à son lancement.

Ce sont en tout cas les blocs que j'ai utilisé dans mon Dockerfile pour le serveur FTP. Voici donc à quoi ressemble ce fichier.

```
root@test:/home/test# cat Dockerfile
FROM stilliard/pure-ftpd

RUN echo "yes" > /etc/pure-ftpd/conf/NoAnonymous
RUN echo "yes" > /etc/pure-ftpd/conf/ChrootEveryone
RUN echo "yes" > /etc/pure-ftpd/conf/PAMAuthentication

EXPOSE 21 30000-30009

CMD /run.sh -c 30 -c 10 -i puredb:/etc/pure-ftpd/pureftpd.pdb
```

Une fois ce fichier créé, on peut construire l'image en précisant le nom de cette-dernière.

```
root@test:/home/test# docker build -t mon-serveur-ftp .
```

Et enfin on lance notre docker run qui va permettre de créer mon container.

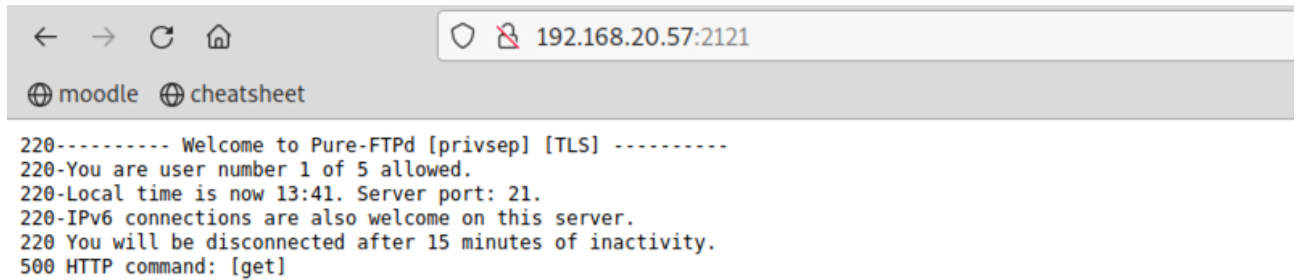
```
root@test:/home/test# docker run -d -p 2121:21 -p 30000-30009:30000-30009 mon-serveur-ftp
```

Dans cette commande on voit le paramètre -d qui va permettre de lancer le container en fond et ainsi de ne pas être directement dedans. Pour y être il faudrait faire un docker exec mais dans mon cas c'est inutile.

Le -p permet de préciser les ports que l'on veut utiliser, 2121 pour la machine hôte et 21 pour le container.

Et on n'oublie pas de préciser le nom de l'image que l'on a précédemment créé.

Une fois que le container a correctement démarré, en tapant l'adresse électronique de ma VM en précisant le bon port, je dois tomber sur mon serveur FTP. Et c'est bien le cas.



```
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 5 allowed.
220-Local time is now 13:41. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
500 HTTP command: [get]
```

Mais après cela fait, j'ai remarqué que quand je redémarrais ma VM ou encore après 15 minutes d'inactivité, le container s'arrêtait tout seul et donc mon serveur était down.

J'ai donc cherché une solution pour que ce container se redémarre tout seul.

La solution se trouve dans la commande docker run. Il existe une option pour préciser cela.

L'option est -- restart=unless-stopped.

Après cette option ajoutée dans mon docker run, j'ai fait le test de reboot ma VM et quand j'ai fait mon docker ps pour lister mes containers, on peut voir qu'il tournait toujours.

SERVEUR DHCP

Pour le serveur DHCP, je n'ai pas réussi à le faire mais j'ai quand même fait des fichiers de configurations que je vais expliquer.

Comme pour le serveur FTP, j'ai créé un Dockerfile que j'ai rempli en utilisant un exemple que j'ai trouvé sur github. Mais comparé au FTP, j'ai également créé un fichier dhcp.conf où je précise à l'intérieur quelques informations comme le nom de mon domaine ou encore le DNS.

Mais l'information la plus importante de ce fichier est la range des adresses que mon DHCP va proposer. Ainsi que le masque que ces adresses auront.

Voici donc ce fichier dhcp.conf

```
GNU nano 6.2          dhcpd.conf
option domain-name "sae.jing.fr"
option domain-name-servers 8.8.8.8, 8.8.4.4;

subnet 192.168.20.0 netmask 255.255.255.0 {
    range 192.168.20.10 192.168.20.199;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.20.255;
    option routers 192.168.20.1;
}

authoritative;
default-lease-time 600;
max-lease-time 7200;

log-facility local7;
```

Et voici également le Dockerfile que j'ai créé.

A screenshot of a terminal window showing a Dockerfile being edited with the GNU nano 6.2 editor. The title bar of the editor shows 'GNU nano 6.2' and 'Dockerfile'. The content of the Dockerfile is as follows:

```
FROM debian:latest

RUN apt update && apt install -y \
    isc-dhcp-server\
    iproute2
RUN apt install sudo

COPY dhcpd.conf /etc/dhcp/dhcpd.conf

EXPOSE 67/udp
EXPOSE 68/udp

CMD ["dhcp", "-f", "-d", "--no-pid"]
```

Ce qu'il ne faut pas oublier quand on essaye de construire l'image, c'est que ces deux fichiers doivent être dans le même dossier pour que ça fonctionne.

Mais malheureusement quand je construisais mon image et que je lançais mon container, rien ne se passait.

J'ai donc bloqué sur ce problème pendant pas mal de jours mais on a utilisé le DHCP de pfSense donc il n'y avait plus besoin d'un serveur DHCP.