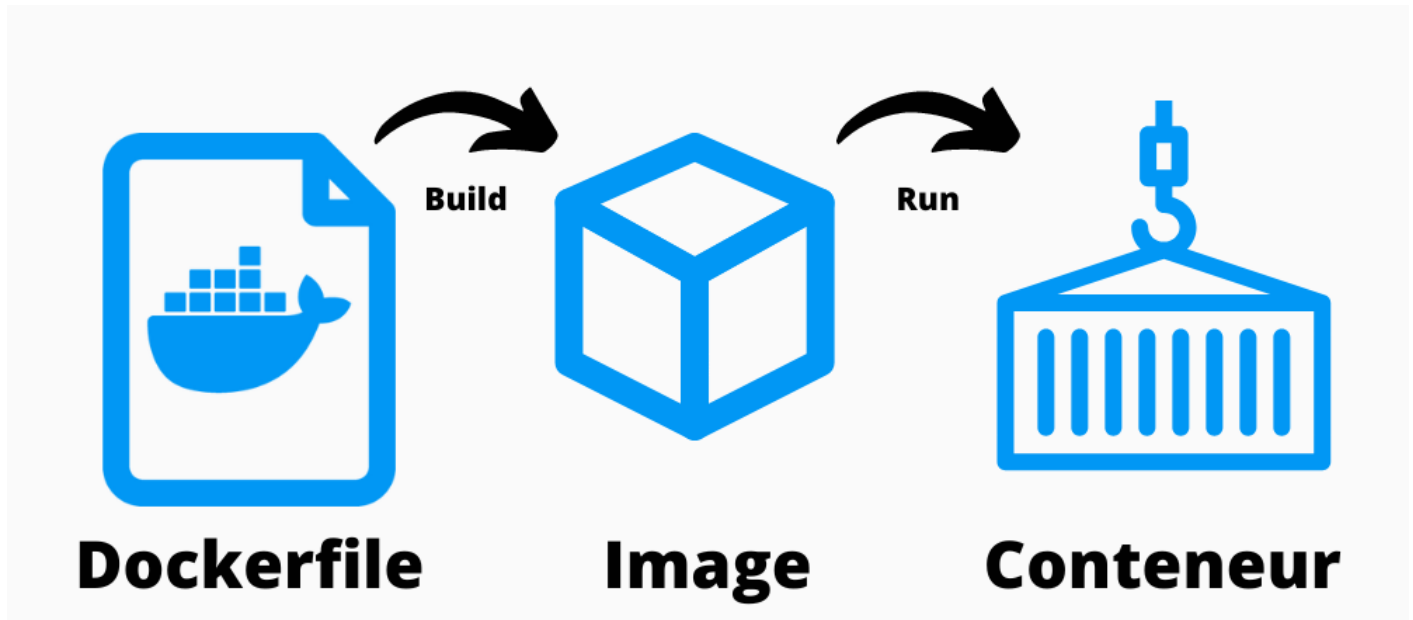


Dockerfile vers Conteneur

- [Créer un Dockerfile](#)
- [Créer un tag de son image](#)
- [Publier son image sur le Docker Hub](#)

Créer un Dockerfile

Process de création d'une image Docker



La création d'un conteneur Docker depuis un Dockerfile se fait en 3 étapes distinct :

- 1) La création du Dockerfile
- 2) La compilation du Dockerfile pour avoir une image Docker
- 3) L'utilisation de l'image Docker pour créer un conteneur

1/ Dockerfile

`RUN`

utilisez l'instruction `RUN` pour exécuter une commande dans votre conteneur.

Limitez au maximum le nombre d'instructions `RUN` , afin de limiter le nombre de layers créées, et donc de réduire la taille de notre image Docker.

`EXPOSE`

L'instruction `EXPOSE` permet d'indiquer le port sur lequel votre application écoute.

`VOLUME`

L'instruction `VOLUME` permet d'indiquer quel répertoire vous voulez partager avec votre host.

`WORKDIR`

Utilisez ensuite l'instruction `WORKDIR` qui permet de modifier le répertoire courant. La commande est équivalente à une commande `cd` en ligne de commande. L'ensemble des commandes qui suivront seront toutes exécutées depuis le répertoire défini.

`ADD`

utilisez l'instruction

`ADD` afin de copier ou de télécharger des fichiers dans l'image. Dans notre cas, nous l'utilisons pour ajouter les **sources de notre application** locale dans le dossier `/app/` de l'image.

```
FROM debian:10

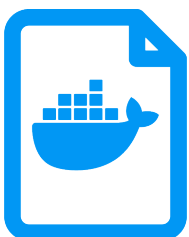
WORKDIR /app
ADD . /app

RUN npm install && npm run build

EXPOSE 3000
VOLUME [/app/logs]

CMD ["npm", "run", "start"]
```

Nous venons de créer un fichier **Dockerfile** qui permettra de créer une image Docker une fois Build.

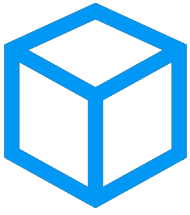


2/ Compiler le Dockerfile

```
docker build --tag monimage .
```

L'argument `-t` permet de **donner un nom à votre image** Docker. Cela permet de retrouver plus facilement votre image par la suite.

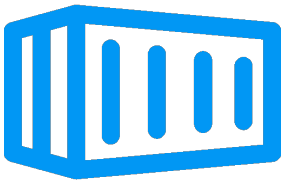
Le `.` est le répertoire où se trouve le Dockerfile ; dans notre cas, à la racine de notre projet.



3/ Utiliser la nouvelle image

```
docker run -d -p 3000:3000 -v /folder/logs:/app/logs --name moncontainer monimage
```

Vous retrouvez, dans le dossier `logs` , les logs de votre application, et vous pourrez y accéder sur le port `3000` , soit via l'URL <http://127.0.0.1:2368>.



En résumé

Pour créer une image Docker, vous savez utiliser les instructions suivantes :

- `FROM` qui vous permet de définir l'image **source** ;
- `RUN` qui vous permet d'exécuter des **commandes** dans votre conteneur ;
- `ADD` qui vous permet **d'ajouter des fichiers** dans votre conteneur ;
- `WORKDIR` qui vous permet de définir votre **répertoire de travail** ;
- `EXPOSE` qui permet de définir les **ports d'écoute** par défaut ;
- `VOLUME` qui permet de définir les **volumes utilisables** ;
- `CMD` qui permet de définir la **commande par défaut** lors de l'exécution de vos conteneurs Docker.

Créer un tag de son image

Il est très utile de créer un tag d'image pour avoir différentes version du code dans des images Docker. Ainsi avoir un historique du projet en image Docker.

Publier son image sur le Docker Hub

Premiers pas avec Docker Hub

Dans un premier temps, inscrivez-vous directement sur le site, c'est gratuit et ça se passe par ici : [Docker Hub - Inscription](#)

Créer le répertoire

Ensuite, sur la page d'accueil, nous allons cliquer sur "**Repository**", puis "**Create repository**".

Un repository Docker Hub va accueillir une seule et unique image docker, que nous allons construire par l'intermédiaire d'un fichier Dockerfile. **Nommez votre repository**, et choisissez l'option "**Public**" avant de cliquer sur "**Create**". Pour ma part, le dépôt se nommera "*checkserv*".

Une fois le compte créé et le répertoire créé il faut se connecter au compte Docker.

```
docker login
```

L'ordinateur est maintenant prêt à manipuler l'image Docker.

Créer un tag

```
docker tag <image>:<tag> <user>/<repository>:<tag>
```

Tag une image référencé par le nom

Pour tag une image local avec le nom "checkserv" dans le répertoire "gpatruno" avec la version "1.0". La commande Docker :

```
docker tag checkserv gpatruno/checkserv:1.0
```

Publier son image sur le Hub

```
docker push <user>/<repository>:<tag>
```

Dans notre cas pour publier l'image Docker sur le Hub il faudra exécuter la commande suivante :

```
docker push gpatruno/checkserv:1.0
```

Ce qui donne le répertoire suivant : [gpatruno/checkserv](https://hub.docker.com/r/gpatruno/checkserv)