

Resume : Dockerisation d'un projet React + Node + Nginx - Bonnes pratiques et erreurs frequentes

1) Bien orthographier les noms d'images

- Exemple d'erreur : FROM ngnix:alpine
- Correction : FROM nginx:alpine

2) Installer les dependances avant de lancer un build

- npm install avant npm run build

3) Copier tout le dossier avant le build

- COPY .. avant RUN npm run build

4) Utiliser le bon dossier de build

- Le build se trouve dans /app/build et non /app/base

5) Respecter l'ordre logique des etapes Dockerfile :

```
FROM node:alpine AS base
WORKDIR /app
COPY package*.json .
RUN npm install
COPY ..
RUN npm run build
FROM nginx:alpine
COPY --from=base /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

6) Ne jamais oublier le fichier .dockerignore

7) Valider ses images et conteneurs apres build

- docker images
- docker ps -a
- docker system prune -a

8) Mapper les ports correctement

- docker run -d -p 8080:80 image

9) Gérer le cache de build en cas de plantage

- docker builder prune

10) Vérifier les chemins et le dossier de travail

- Utiliser WORKDIR /app et les chemins relatifs

Résumé Express :

Erreur -> Solution

- Mauvais nom image -> Vérifie sur Docker Hub
- Pas d'install -> npm install avant build
- Copier partiel -> COPY .. avant build
- Mauvais dossier cible -> /app/build
- Ordre anarchique -> Respecter la séquence
- Trop de fichiers -> .dockerignore
- Ports mal mappés -> -p machine:conteneur
- Cache bloqué -> docker builder prune
- Pas de contrôle après build -> docker images / ps -a