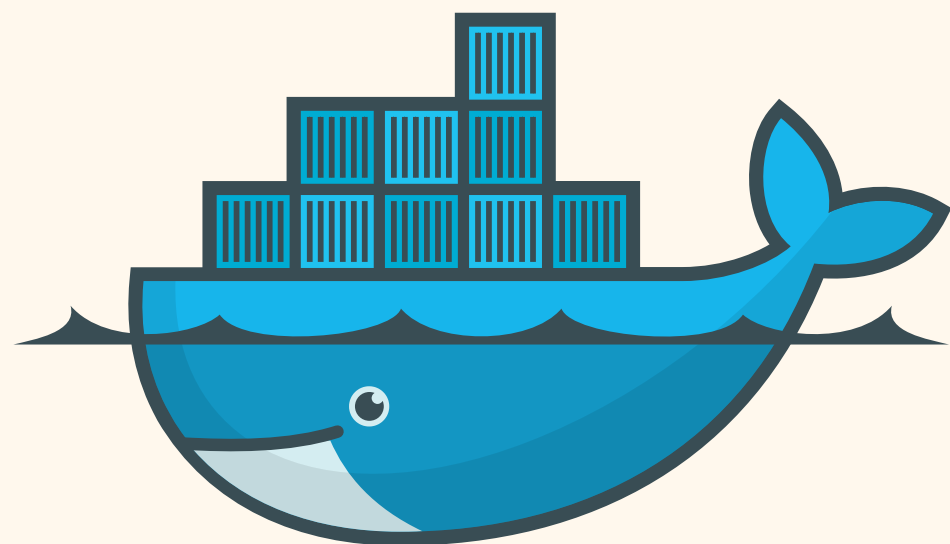
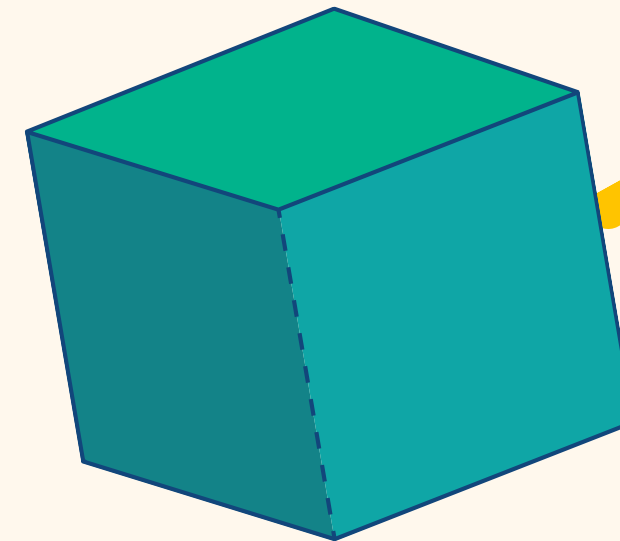


Docker



Découverte d'un truc exceptionnel

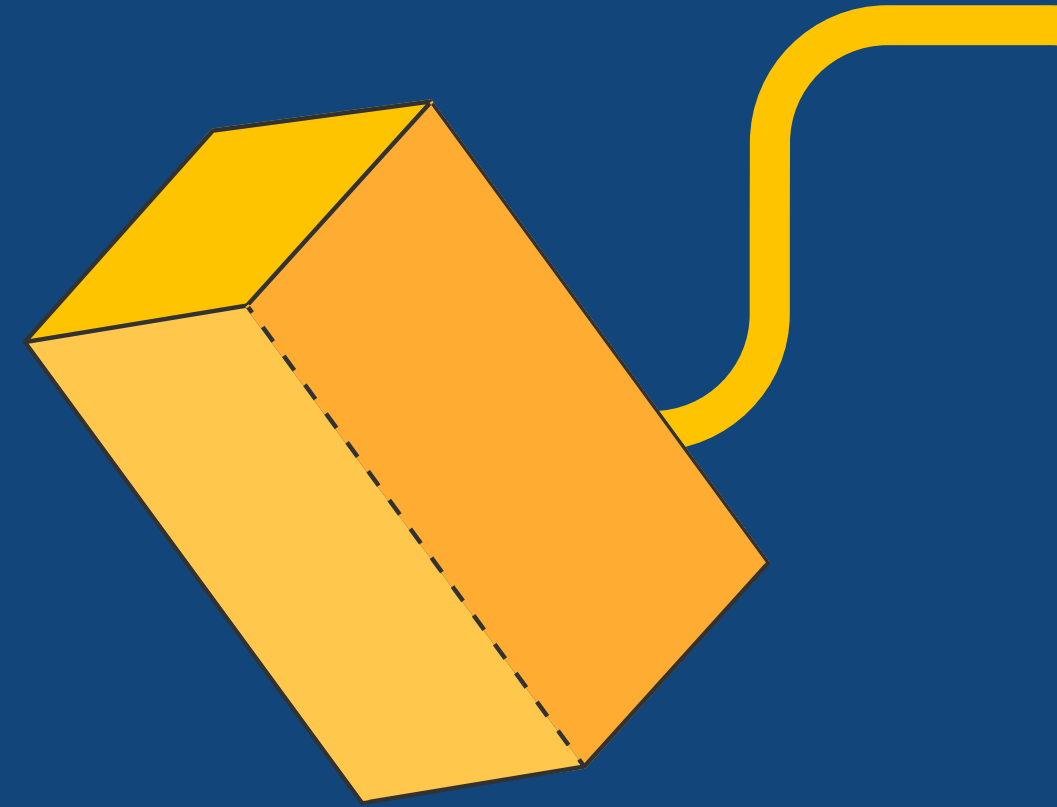


Manu-cj



Qu'est-ce que Docker ?

- Docker est un outil de virtualisation léger, qui permet de créer des conteneurs.
- Un **conteneur** est une boîte qui regroupe une application et ses dépendances, permettant de l'exécuter de manière fiable sur n'importe quelle machine.
-
- Docker permet d'éviter le problème du "**ça marche sur ma machine**" en garantissant que l'application fonctionne de la même manière, quelle que soit l'environnement.





Pourquoi Docker ?



Portabilité	Crée un environnement de développement isolé, facilement transportable.
Isolation	Chaque application et ses dépendances sont isolées dans des conteneurs, ce qui empêche les conflits.
Efficacité	Moins de surcharges par rapport aux machines virtuelles classiques.



Dockerfile ?

Dockefile

```
FROM node:16  
WORKDIR /app  
COPY package*.json ./  
RUN npm install  
COPY . .  
EXPOSE 3000  
CMD ["npm", "start"]
```

Manu Crj

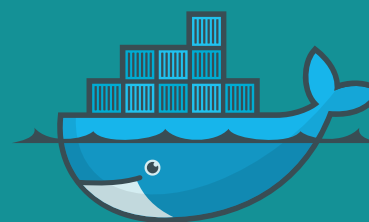
- Le **Dockerfile** est un fichier texte contenant une série d'instructions qui décrivent comment construire une image Docker.
- Chaque ligne dans le **Dockerfile** représente une étape de la construction de l'image, telle que l'installation d'une application ou la configuration d'un environnement.





Manu-cj

```
kerfile > ...  
# Utiliser une image officielle de Node.js comme base  
FROM node:16  
  
# Créer un répertoire pour l'application  
WORKDIR /app  
  
# Copier les fichiers package.json et package-lock.json  
COPY package*.json ./  
  
# Installer les dépendances  
RUN npm install  
  
# Copier le reste du code de l'application  
COPY . .  
  
# Exposer le port sur lequel l'application va tourner  
EXPOSE 3000  
  
# Lancer l'application  
CMD ["npm", "start"]
```





Docker Compose ?

Docker-compose

Trop de choses à noter

Manu Crj

- **Docker Compose** est un outil qui permet de définir et d'exécuter des applications multi-conteneurs. Au lieu de lancer plusieurs conteneurs avec des commandes docker run, on utilise un fichier docker-compose.yml pour définir et gérer plusieurs services.





Manu-cj

```
docker-compose.yml
1  version: '3.7'
2
3  >Run All Services
4  services:
5    >Run Service
6    app:
7      build: .
8      ports:
9        - "3000:3000" # Expose le port 3000 de l'application sur le port 3000 de l'hôte
10     depends_on:
11       - db # Dépendance au service "db" (la base de données)
12     environment:
13       - DB_HOST=${DB_HOST} # Nom d'hôte de la base de données
14       - DB_USER=${DB_USER} # Utilisateur de la base de données
15       - DB_PASSWORD=${DB_PASSWORD} # Mot de passe de la base de données
16       - DB_NAME=${DB_NAME} # Nom de la base de données
17
18   >Run Service
19   db:
20     image: mysql:5.7 # Image Docker de MySQL version 5.7
21     environment:
22       MYSQL_ROOT_PASSWORD: example # Mot de passe root pour MySQL
23       MYSQL_DATABASE: mydb # Nom de la base de données à créer
24     volumes:
25       - mysql-data:/var/lib/mysql # Volume pour persister les données MySQL
26     ports:
27       - "3306:3306" # Expose le port 3306 de MySQL sur le port 3306 de l'hôte
28
29   >Run Service
30   phpmyadmin:
31     image: phpmyadmin/phpmyadmin # Image Docker de phpMyAdmin
32     environment:
33       PMA_HOST: db # Nom d'hôte de la base de données
34       MYSQL_ROOT_PASSWORD: example # Mot de passe root pour accéder à MySQL
35     ports:
36       - "8080:80" # Expose le port 80 de phpMyAdmin sur le port 8080 de l'hôte
37     depends_on:
38       - db # Dépendance au service "db" (la base de données)
39
40   volumes:
41     mysql-data:
42       driver: local # utilisation d'un volume local pour stocker les données
```





Manu-cj

Comment Tester l'Application?

```
manu@Bachelor:~/Code/test$ docker-compose up --build
```

- Construire l'image Docker pour l'application Node.js (si elle n'existe pas déjà).
- Démarrer les services définis dans docker-compose.yml (l'application, la base de données et phpMyAdmin).





Conclusion

Résumé :

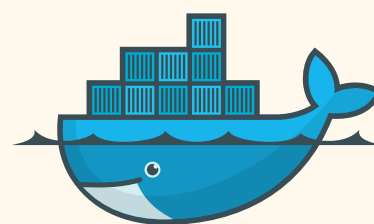
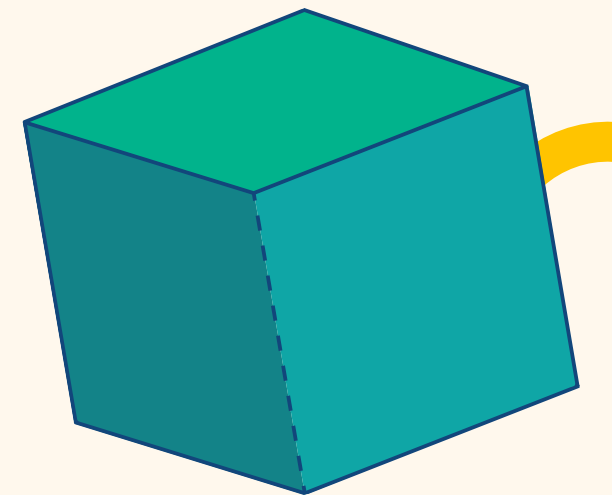
- Docker permet de créer des environnements isolés, ce qui garantit que ton application fonctionne partout, tout le temps. Le Dockerfile est un fichier qui décrit comment créer une image Docker, tandis que docker-compose.yml permet de gérer des applications avec plusieurs services.

Pourquoi l'utiliser ?

- Docker simplifie le déploiement et la gestion des applications, surtout lorsqu'elles dépendent de plusieurs services (comme une base de données). Il assure aussi que les développeurs et les équipes peuvent travailler dans des environnements cohérents.

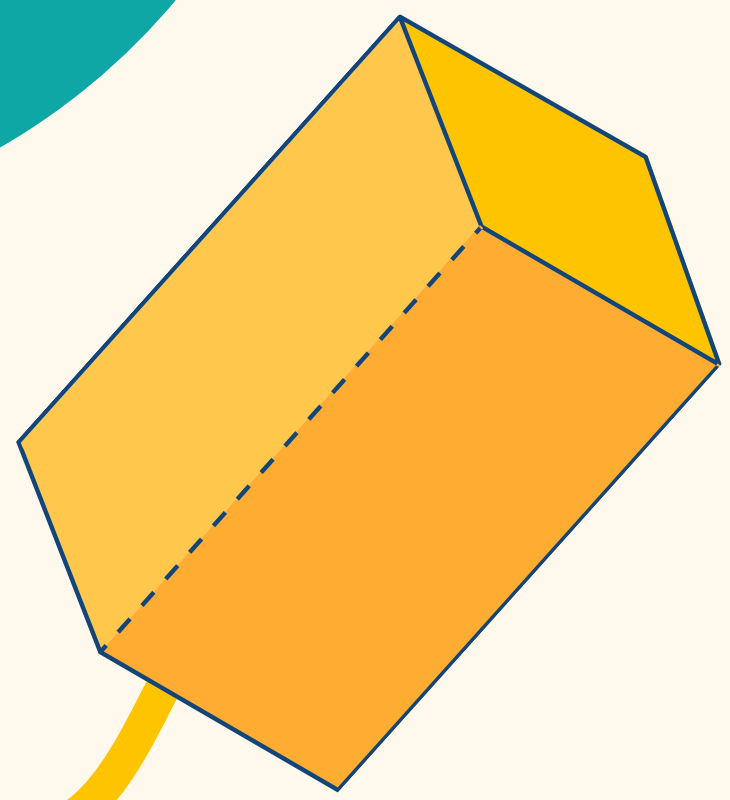
Prochaines étapes :

- Explore les différentes images Docker disponibles pour d'autres services, comme Redis, MongoDB, ou Nginx, et apprends à les intégrer dans ton propre projet.





Manu-cj





Manu
manu-cj

Follow

