

Chapitre 04 : TD Lancement d'une application avec docker

LPRGL3

PROJET INFORMATIQUE II



docker

Intérêt du chapitre

- Ce TD vous permettra d'acquérir des compétences pratiques et utiles pour:
 - Développer et déployer des applications plus rapidement et facilement.
 - Améliorer la portabilité et la scalabilité de vos applications.
 - Gérer vos applications de manière plus efficace et fiable.

Objectifs

- Définir les concepts clés de Docker et ses avantages.
- Installer et configurer Docker sur votre machine.
- Créer et utiliser des images Docker pour vos applications.
- Démarrer, gérer et surveiller des conteneurs.
- Explorer les options de déploiement en production.

Prérequis

- Logiciels:
 - Docker Desktop.
 - Lien pour télécharger Docker Desktop:
<https://docs.docker.com/get-docker/>
 - Un éditeur de texte

Sommaire

- Obtenir Docker
- Présentation de Docker
- Qu'est-ce qu'un conteneur ?
- Exécuter un conteneur
- Exécuter des images Docker Hub
- Quizz
- Applications multi-conteneurs
- Rendre persistantes les données d'un conteneur
- Accéder à un dossier local
- Conteneurisez votre application
- Publiez votre image sur Docker Hub
- Quizz

Obtenir Docker

- Docker est disponible pour Windows, Mac et Linux.
- Nous utiliserons Docker Desktop.
- Lien pour télécharger Docker Desktop:
<https://docs.docker.com/get-docker/> .
- Autres options d'installation:
 - Docker Engine.
 - Vous pouvez utiliser Docker dans un environnement cloud.

Présentation de Docker

- Docker est une plateforme pour simplifier le développement et le déploiement d'applications.

Qu'est-ce qu'un conteneur ?

- Un conteneur est un environnement isolé pour votre code.
- Il s'exécute sur l'environnement qui vous est fourni par Docker Desktop.
- Les conteneurs possèdent tout ce dont votre code a besoin pour s'exécuter, jusqu'à un système d'exploitation de base.

Qu'est-ce qu'un conteneur ?

- Vous pouvez utiliser Docker Desktop pour gérer et explorer vos conteneurs.


Qu'est-ce qu'un conteneur ?

Étape 1 : Configurer le guide pratique

- La première chose dont vous avez besoin est un conteneur en cours d'exécution.
- Utilisez les instructions suivantes pour exécuter un conteneur :
 1. Ouvrez Docker Desktop et cliquez sur la barre de recherche.
 2. Indiquez `docker/welcome-to-docker` dans la recherche et sélectionnez "Exécuter" ("Run").

Qu'est-ce qu'un conteneur ?

Étape 1 : Configurer le guide pratique

 **Run a new container**
docker/welcome-to-docker:latest

Optional settings ^

Container name
welcome-to-docker

A random name is generated if you do not provide one.

Ports
Enter "0" to assign randomly generated host ports.

Host port
8088 :80/tcp

Volumes

Host path ... Container path +

Environment variables

Variable Value +

Cancel

Run

Qu'est-ce qu'un conteneur ?

Étape 1 : Configurer le guide pratique

- 3. Développez les "Paramètres optionnels" ("Optional settings").
- 4. Dans la section "Nom du conteneur" ("Container name"), spécifiez `welcome-to-docker`.
- 5. Dans la section "Port hôte" ("Host port"), indiquez `8088`.
- 6. Cliquez sur "Exécuter" ("Run").

Qu'est-ce qu'un conteneur ?




Étape 2 : Visualiser les conteneurs dans Docker Desktop

- Vous venez d'exécuter un conteneur !
- Vous pouvez le voir dans l'onglet "Conteneurs" de Docker Desktop.
- Ce conteneur exécute un simple serveur web qui affiche un site web basique.

Qu'est-ce qu'un conteneur ?

Étape 3 : Visualiser le frontend






- Le frontend est accessible sur le port 8088 de votre machine locale (localhost).
- Cliquez sur le lien dans la colonne "Port(s)" de votre conteneur, ou visitez <http://localhost:8088> dans votre navigateur pour l'afficher.

<input type="checkbox"/>	Name ↓	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	welcome-to-docker 7d8bd512b269	docker/wel	Running	0%	8088:80	http://localhost:8088	  

Qu'est-ce qu'un conteneur ?

Étape 4 : Explorer votre conteneur





- Docker Desktop vous permet de visualiser et d'interagir facilement avec différents aspects de votre conteneur.
- Sélectionnez votre conteneur, puis cliquez sur "Fichiers" ("Files") pour explorer le système de fichiers isolé de votre conteneur.

<input type="checkbox"/>	Name ↓	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 welcome-to-docker 7d8bd512b269	docker/wel	Running	0%	8088:80	12 minutes ago	   

Qu'est-ce qu'un conteneur ?

Étape 5 : Arrêter votre conteneur

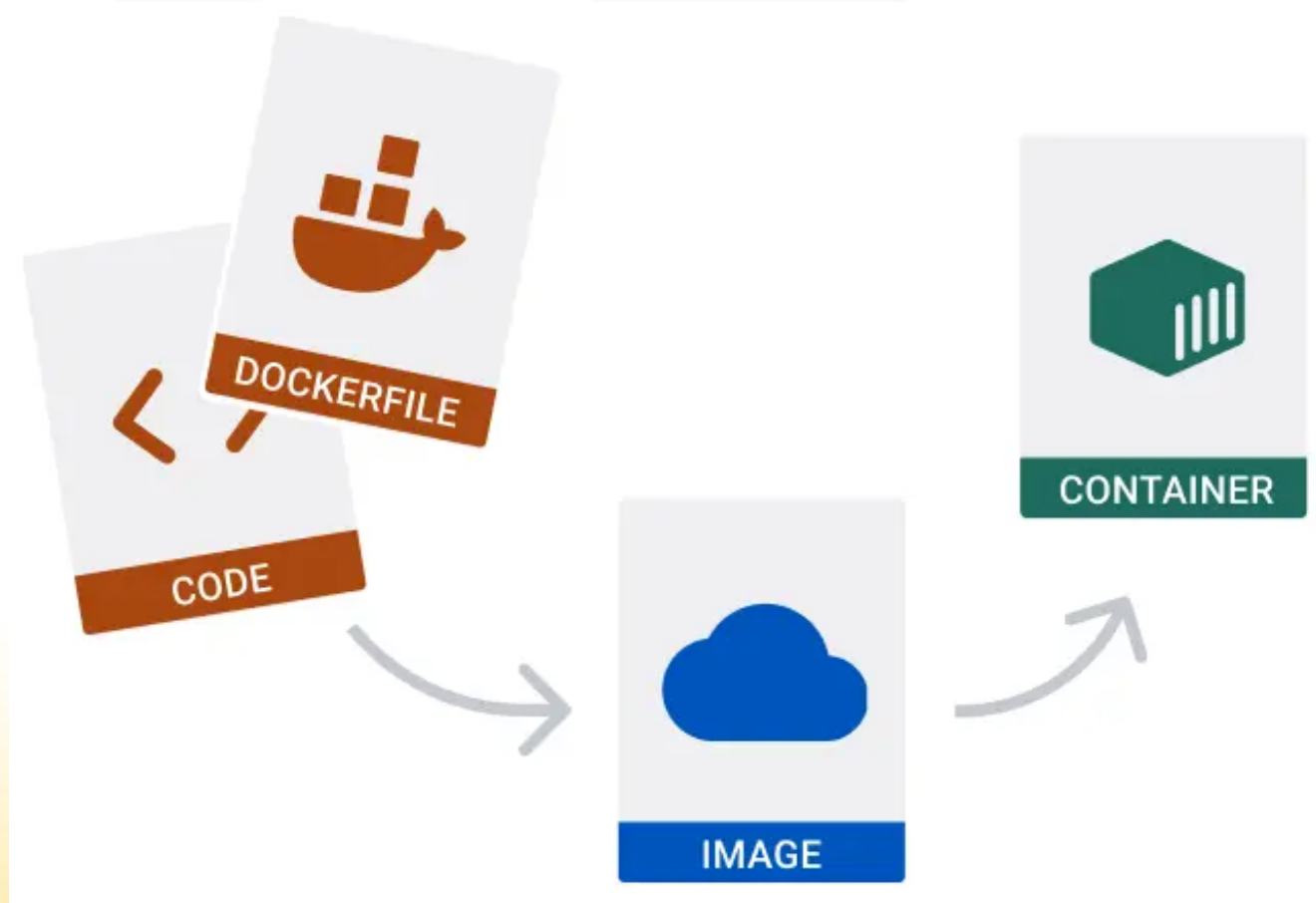
- Le conteneur "welcome-to-docker" continue de fonctionner jusqu'à ce que vous l'arrêtiez.
- Pour arrêter le conteneur dans Docker Desktop, allez dans l'onglet "Conteneurs" et cliquez sur l'icône "Stop" dans la colonne "Actions" de votre conteneur.

<input type="checkbox"/>	Name ↑	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 welcome-to-docker 7d8bd512b269	docker/wel	Running	0%	8088:80	8 minutes ago	  

Comment exécuter un conteneur?

- Dans cette partie, vous apprendrez les étapes de base pour construire une image et exécuter votre propre conteneur.
- Nous utiliserons un exemple d'application Node.js, mais il n'est pas nécessaire de connaître Node.js.

Comment exécuter un conteneur?



Comment exécuter un conteneur ?

Étape 1 : Obtenir l'application d'exemple

- Vous pouvez cloner le dépôt de l'application exemple ou simplement utiliser le fichier joint.
- Cloner avec git: Ouvrez une fenêtre de terminal et utilisez la commande suivante:
 - `git clone https://github.com/docker/welcome-to-docker`



welcome-to-docker-main.zip

Comment exécuter un conteneur ?

Étape 2 : Visualiser le Dockerfile dans votre dossier de projet

- Pour exécuter votre code dans un conteneur, l'élément fondamental est un **Dockerfile**.
- **Il décrit ce qui est contenu dans un conteneur.**
- Cet exemple contient déjà un Dockerfile.
- Pour vos projets personnels, vous devrez créer votre propre Dockerfile.

Comment exécuter un conteneur ?

Étape 2 : Visualiser le Dockerfile dans votre dossier de projet

- Ouvrez le Dockerfile dans un éditeur de code ou de texte et explorez son contenu.

Comment exécuter un conteneur ?

Étape 3 : Construire votre première image

- Vous avez toujours besoin d'une image pour exécuter un conteneur.
- Ouvrez un terminal et exécutez les commandes suivantes pour construire l'image.
- Remplacez `/chemin/vers/welcome-to-docker/` par le chemin vers votre dossier :
 - `cd /chemin/vers/welcome-to-docker/`
 - `docker build -t welcome-to-docker .`

Comment exécuter un conteneur ?

Étape 3 : Construire votre première image

- La construction de l'image peut prendre un certain temps.
- Une fois votre image construite, vous pouvez la voir dans l'onglet "Images" de Docker Desktop.

Comment exécuter un conteneur ?

Étape 4 : Exécuter votre conteneur

- Pour exécuter votre image en tant que conteneur :
- 1. Dans Docker Desktop, allez dans l'onglet "Images".
- 2. À côté de votre image, cliquez sur "Exécuter" ("Run").
- 3. Développez les "Paramètres optionnels" ("Optional settings").

Comment exécuter un conteneur ?

Étape 4 : Exécuter votre conteneur

- 4. Dans la section "Port hôte" ("Host port"), indiquez `8089`.
- 5. Cliquez sur "Exécuter" ("Run").

Comment exécuter un conteneur ?

Étape 5 : Visualiser le frontend

- Vous pouvez accéder à votre conteneur en cours d'exécution via Docker Desktop.
- Cliquez sur le lien à côté de votre conteneur ou visitez <http://localhost:8089> pour visualiser le frontend.

Comment exécuter un conteneur ?

Résumé

- Vous avez construit votre propre image et l'avez exécutée en tant que conteneur.
- En plus de cela, vous pouvez également exécuter des images provenant de Docker Hub.

Exécuter des images Docker Hub

- Comment exécuter des images préconfigurées à partir de Docker Hub, le registre public d'images Docker?

Exécuter des images Docker Hub

Étape 1 : Rechercher une image

- 1. Ouvrez Docker Desktop.
- 2. Dans la barre de recherche, saisissez le nom de l'image que vous souhaitez exécuter.
- Par exemple, pour exécuter une image Node.js, saisissez `node`.
- 3. Parcourez les résultats de la recherche et sélectionnez l'image qui vous convient.


Exécuter des images Docker Hub

Étape 2 : Exécuter l'image

- 1. Cliquez sur le bouton "Exécuter" ("Run") à côté de l'image sélectionnée.
- 2. Dans la fenêtre "Exécuter l'image" ("Run Image"), vous pouvez :
 - Modifier le nom du conteneur(facultatif).
 - Définir les ports à exposer(facultatif).
 - Ajouter des variables d'environnement(facultatif).
- 3. Cliquez sur "Exécuter" ("Run").

Exécuter des images Docker Hub

Étape 2 : Exécuter l'image

**Run a new container**
docker/welcome-to-docker:latest

Optional settings ^

A random name is generated if you do not provide one.

Ports
Enter "0" to assign randomly generated host ports.

Host port

:80/tcp

Volumes

...

+

Environment variables

+

Cancel

Run

Exécuter des images Docker Hub

Étape 3 : Explorer le conteneur

- Une fois le conteneur démarré, vous pouvez l'explorer de différentes manières :
 - Afficher les journaux du conteneur : sélectionnez le conteneur dans l'onglet "Conteneurs" ("Containers") et cliquez sur "Journaux" ("Logs").
 - Accéder au terminal du conteneur : sélectionnez le conteneur et cliquez sur "Terminal" ("Terminal").

Exécuter des images Docker Hub

Étape 3 : Explorer le conteneur

- Afficher les ports exposés : sélectionnez le conteneur et regardez la section "Ports" ("Ports") dans l'onglet "Détails" ("Details").

Containers
[Give feedback](#)

Container CPU usage ⓘ

0.00% / 1000% (10 cores available)


Container memory usage ⓘ

9.53MB / 7.46GB

[Show charts](#)

☰

☒ Only show running containers

<input type="checkbox"/>	Name ↑	Image	Status	CPU (%)	Port(s)	Actions
<input type="checkbox"/>	<div>  magical_chatelet 5c77b5a4c1e4 </div>	docker/wel	Running	0%	8090:80	<div> <div>■</div> <div>⋮</div> <div>🗑️</div> </div>

Exécuter des images Docker Hub

Étape 4 : Arrêter le conteneur

- Lorsque vous avez terminé avec le conteneur, vous pouvez l'arrêter de différentes manières :
 - Dans Docker Desktop : sélectionnez le conteneur et cliquez sur "Arrêter" ("Stop").
 - Dans le terminal : utilisez la commande ``docker stop <nom_du_conteneur>``.

Exécuter des images Docker Hub

En résumé

- Exécuter des images Docker Hub est un moyen simple et rapide de lancer des applications préconfigurées dans des conteneurs.

Exécuter des applications multi-conteneurs

- Après le chapitre "Comment exécuter un conteneur ?", vous savez qu'il faut démarrer chaque conteneur individuellement.
- Imaginez à quel point ce serait pratique si un outil permettait de démarrer plusieurs conteneurs avec une seule commande.
- Cet outil, c'est **Docker Compose**.

Exécuter des applications multi-conteneurs

Étape 1 : Obtenir l'application d'exemple



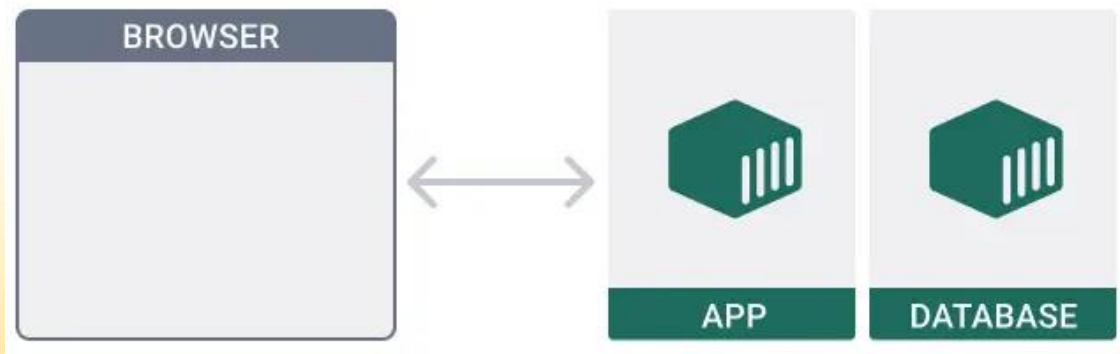
multi-container-app-main.zip

- Vous pouvez également cloner le dépôt de l'application exemple sur GitHub.
 - git clone <https://github.com/docker/multi-container-app>
- L'application d'exemple est une simple application "todo" (liste de tâches) construite avec ExpressJS et Node.js.

Exécuter des applications multi-conteneurs

Étape 1 : Obtenir l'application d'exemple

- Elle stocke toutes les tâches dans une base de données MongoDB.
- Vous n'avez pas besoin de connaître ces technologies pour continuer ce chapitre.
- L'architecture de l'application d'exemple



Exécuter des applications multi-conteneurs

Étape 2 : Analyser le fichier compose

- Observez les fichiers de l'application.
- Notez qu'elle comporte un fichier **compose.yaml**.
- Ce fichier indique à Docker comment exécuter votre application.
- Ouvrez-le dans un éditeur de code ou de texte pour voir son contenu.

Exécuter des applications multi-conteneurs

Étape 3 : Exécuter l'application

- Pour exécuter cette application multi-conteneur, ouvrez un terminal et exécutez les commandes suivantes.
- Remplacez `/chemin/vers/application-multi-conteneur` par le chemin d'accès au répertoire de votre application.
 - `cd /chemin/vers/application-multi-conteneur`
 - `docker compose up -d`

Exécuter des applications multi-conteneurs

Étape 3 : Exécuter l'application

- Dans la commande précédente, l'option -d indique à Docker Compose de s'exécuter en mode détaché (en arrière-plan).

Exécuter des applications multi-conteneurs

Étape 4 : Voir l'interface et ajouter des tâches

- Dans l'onglet "Containers" de Docker Desktop, vous devriez maintenant avoir une pile d'applications avec deux conteneurs en cours d'exécution (todo-app et todo-database).
- Pour voir l'interface de l'application :
 - Dans Docker Desktop, développez la pile d'applications dans "Containers".
 - Cliquez sur le lien vers le port 3000 dans la colonne "Port(s)" ou allez sur <http://localhost:3000>.

Exécuter des applications multi-conteneurs

Étape 4 : Voir l'interface et ajouter des tâches

- Ajoutez quelques tâches dans l'interface, puis ouvrez <http://localhost:3000> dans un nouvel onglet du navigateur. Vous verrez que les tâches sont toujours présentes.

Exécuter des applications multi-conteneurs

Étape 5 : Développer au sein de vos conteneurs

- Lorsque vous développez avec Docker, il se peut que vous ayez besoin de mettre à jour et de prévisualiser automatiquement vos services en cours d'exécution lorsque vous modifiez et sauvegardez votre code.
- Pour cela, vous pouvez utiliser **Docker Compose Watch**.
- Pour exécuter Compose Watch et voir les changements en temps réel :

Exécuter des applications multi-conteneurs

Étape 5 : Développer au sein de vos conteneurs

- 1. Ouvrez un terminal et accédez au répertoire de votre application :
 - `cd /chemin/vers/application-multi-conteneur/`
- 2. Exécutez la commande Docker Compose Watch :
 - `docker compose watch`
- 3. Modifiez le fichier `app/views/todos.ejs` :
 - Ouvrez le fichier dans un éditeur de texte ou de code, puis modifiez le texte à la ligne 18.

Exécuter des applications multi-conteneurs





Étape 5 : Développer au sein de vos conteneurs

- 4. Enregistrez les modifications.
- 5. Visualisez votre application pour voir les changements :
 - Accédez à votre application via `http://localhost:3000` pour voir les modifications reflétées en temps réel.

Exécuter des applications multi-conteneurs

Étape 6 : Tout supprimer et recommencer

- Avoir votre configuration stockée dans un fichier Compose a un autre avantage : vous pouvez facilement tout supprimer et recommencer à zéro.

<input type="checkbox"/>	Name ↓	Image	Status	CPU (%)	Port(s)	Actions
<input type="checkbox"/>	>  multi-container-app		Running (2/2)	0.43%		<div>    </div> <div>Delete</div>

Exécuter des applications multi-conteneurs

Étape 6 : Tout supprimer et recommencer

- Pour supprimer la pile d'applications :
 - Ouvrez l'onglet "Containers" de Docker Desktop
 - Sélectionnez l'icône de suppression à côté de votre pile d'applications.

Quizz

- 1. Quel est l'outil principal utilisé pour gérer et explorer les conteneurs Docker sur Windows et MacOS ?
 - A. Git
 - B. Le terminal
 - C. Docker Desktop
 - D. Visual Studio Code

Quizz

- 2. Qu'est-ce qui est utilisé pour décrire ce qui est contenu dans un conteneur Docker ?
- A. Le code source de l'application
- B. Les dépendances de l'application
- C. Le système d'exploitation
- D. Un Dockerfile

Quizz

- 3. Comment pouvez-vous visualiser le frontend d'une application en cours d'exécution dans un conteneur ?
- A. En utilisant le terminal
- B. En se connectant au terminal du conteneur
- C. En cliquant sur le lien dans la colonne "Port(s)" de Docker Desktop
- D. En utilisant un navigateur web

Quizz

- 4. Quelle est la commande utilisée pour construire une image Docker à partir d'un Dockerfile ?
- A. docker run
- B. docker pull
- C. docker push
- D. docker build

Quizz

- 5. Où pouvez-vous trouver des images Docker préconfigurées à exécuter ?
- A. Dans le terminal
- B. Dans Visual Studio Code
- C. Sur Docker Hub
- D. Dans le dossier de votre projet

Quizz

- 6. Quelle est la différence entre une image Docker et un conteneur Docker ?
 - A. Une image est un instantané statique, tandis qu'un conteneur est une instance en cours d'exécution de l'image.
 - B. Une image est plus grande qu'un conteneur.
 - C. Un conteneur est plus rapide qu'une image.
 - D. Une image est un fichier, tandis qu'un conteneur est un processus.

Quizz

- 7. Que signifie "exécuter un conteneur en mode détaché" ?
 - A. Le conteneur s'exécute en arrière-plan, même après la fermeture de la fenêtre du terminal.
 - B. Le conteneur est accessible uniquement depuis le réseau local.
 - C. Le conteneur est isolé des autres conteneurs.
 - D. Le conteneur n'a pas accès au système d'exploitation hôte.

Quizz

- 8. Comment pouvez-vous accéder aux journaux d'un conteneur en cours d'exécution?
- A. En utilisant le terminal
- B. En se connectant au terminal du conteneur
- C. En utilisant l'interface graphique de Docker Desktop
- D. Toutes les réponses ci-dessus.

Quizz

- 9. Quelles sont les bonnes pratiques pour sécuriser les conteneurs Docker ?
- A. Exécuter tous les conteneurs avec l'accès root.
- B. Utiliser des images de confiance.
- C. Minimiser les privilèges.
- D. B et C

Quizz

- 10. Qu'est-ce que Docker Compose ?
- A. Un outil pour créer des images Docker
- B. Un outil pour gérer les conteneurs Docker
- C. Un outil pour orchestrer des applications multi-conteneurs
- D. Un outil pour partager des images Docker

Persistance des données dans les conteneurs

- Ce chapitre vous explique comment assurer la persistance des données entre plusieurs conteneurs.
- Docker isole tout le contenu d'un conteneur (code, données) de votre système de fichiers local.
- Par conséquent, **lorsque vous supprimez un conteneur, son contenu est intégralement détruit.**

Persistance des données dans les conteneurs

- Parfois, il est nécessaire de conserver les données générées par un conteneur. Pour cela, vous pouvez utiliser des volumes.



Persistance des données dans les conteneurs

Étape 1 : Obtenir l'application d'exemple



multi-container-app-main.zip

- Vous pouvez cloner le dépôt de l'application en utilisant git:
 - `git clone https://github.com/docker/multi-container-app`

Persistance des données dans les conteneurs

Étape 2: Ajouter un volume pour la persistance des données

- **Les volumes permettent de conserver les données même après la suppression d'un conteneur.**
- **Il s'agit d'emplacements sur votre système de fichiers, gérés automatiquement par Docker Desktop.**
- Pour ajouter un volume à ce projet :
- 1. Ouvrez le fichier `compose.yaml` dans un éditeur de texte.

Persistance des données dans les conteneurs

Étape 2: Ajouter un volume pour la persistance des données

- 2. Décommentez les lignes suivantes :
- yamll
 - todo-database:
 - # ...
 - volumes:
 - - database:/data/db
 - # ...
 - volumes:
 - database:

Persistance des données dans les conteneurs

Étape 2: Ajouter un volume pour la persistance des données

- Cette configuration indique à Compose de monter le volume nommé `database` à l'emplacement `/data/db` dans le conteneur du service `todo-database`.

Persistance des données dans les conteneurs

Étape 3 : Exécuter l'application

- Exécutez les commandes suivantes dans un terminal (en remplaçant le chemin) pour lancer l'application multi-conteneur :
 - `cd /chemin/vers/application-multi-conteneur/`
 - `docker compose up -d`

Persistance des données dans les conteneurs

Étape 4 : Voir l'interface et ajouter des tâches

- Dans Docker Desktop (onglet "Containers"), vous devriez avoir une pile d'applications avec deux conteneurs en cours d'exécution.
- Pour voir l'interface et ajouter des tâches :
 - Dans Docker Desktop, développez la pile d'applications.
 - Cliquez sur le lien vers le port 3000 ou ouvrez <http://localhost:3000>.
 - Ajoutez quelques tâches dans l'interface.

Persistence des données dans les conteneurs

Étape 5 : Supprimer la pile d'applications et relancer les conteneurs

- Peu importe le nombre de fois que vous supprimez et recréez les conteneurs, Docker Desktop conservera vos données et les rendra accessibles via le volume `database`.
- Pour supprimer la pile d'applications :
 - Ouvrez l'onglet "Containers" dans Docker Desktop.
 - Sélectionnez l'icône de suppression à côté de votre pile d'applications.

Persistence des données dans les conteneurs

Étape 5 : Supprimer la pile d'applications et relancer les conteneurs

- Relancez l'application en suivant l'étape 3. Notez que les tâches que vous aviez créées sont toujours présentes grâce à la persistance du volume.

Persistance des données dans les conteneurs

Résumé

- Dans ce chapitre, vous avez appris à faire persister des données entre conteneurs à l'aide de volumes.
- Cela est utile pour conserver et partager des données entre conteneurs isolés et éphémères.

Accéder à un dossier local depuis un conteneur

- Ce chapitre explique comment accéder à un dossier de votre ordinateur depuis un conteneur Docker.
- Par défaut, les conteneurs sont isolés de votre système de fichiers local.
- Ils ne peuvent pas accéder directement à vos dossiers.

Accéder à un dossier local depuis un conteneur

- Parfois, il est nécessaire de partager un dossier de votre ordinateur avec un conteneur.
- Pour cela, vous pouvez utiliser des **montages de liens** ("bind mounts").

Accéder à un dossier local depuis un conteneur

Étape 1 : Obtenir l'application d'exemple



bindmount-apps-main.zip

- Vous pouvez cloner le dépôt de l'application :
 - git clone <https://github.com/docker/bindmount-apps>

Accéder à un dossier local depuis un conteneur

Étape 2 : Ajouter un montage de lien dans Compose

- **Un montage de lien permet de partager un dossier de votre ordinateur avec un conteneur.**
- Pour ajouter un montage de lien au projet :
- 1. Ouvrez le fichier `compose.yaml` dans un éditeur de texte.
- 2. Décommentez les lignes suivantes :

Accéder à un dossier local depuis un conteneur

Étape 2 : Ajouter un montage de lien dans Compose

- yamll
 - todo-app:
 - # ...
 - volumes:
 - - ./app:/usr/src/app
 - - /usr/src/app/node_modules

Accéder à un dossier local depuis un conteneur

Étape 2 : Ajouter un montage de lien dans Compose

- Cette configuration indique à Compose :
 - De monter le dossier local `./app` dans le conteneur, à l'emplacement `/usr/src/app`. Cela permet de remplacer le contenu statique du dossier `/usr/src/app` du conteneur et de créer un conteneur dédié au développement.
 - De ne pas écraser le répertoire `node_modules` du conteneur pour préserver les paquets installés dans celui-ci.

Accéder à un dossier local depuis un conteneur

Étape 3 : Exécuter l'application

- Exécutez les commandes suivantes dans un terminal (en remplaçant le chemin) pour lancer votre application :
 - `cd /chemin/vers/bindmount-apps/`
 - `docker compose up -d`

Accéder à un dossier local depuis un conteneur

Étape 4 : Développer l'application

- Utilisez les outils du conteneur tout en développant l'application sur votre ordinateur.
- Les changements apportés localement seront répercutés dans le conteneur.
- Pour tester :
 1. Modifiez le fichier ``app/views/todos.ejs`` sur votre ordinateur (la chaîne "Please add some task.")

Accéder à un dossier local depuis un conteneur

Étape 4 : Développer l'application

- 2. Enregistrez le fichier, puis accédez ou rechargez <http://localhost:3001> pour voir les changements.

Accéder à un dossier local depuis un conteneur

Résumé

- Dans ce guide, vous avez appris à partager un dossier local avec un conteneur.
- Cela vous permet de développer plus rapidement, en évitant de reconstruire votre conteneur à chaque modification du code.

Conteneurisez votre application

- Lorsque vous travaillez avec des conteneurs, vous devez généralement créer un Dockerfile (pour définir votre image) et un fichier `compose.yaml` (pour configurer l'exécution).
- Pour vous aider, Docker Desktop dispose de la commande **`docker init`**.
- Exécutée dans le dossier de votre projet, elle génère les fichiers nécessaires à la conteneurisation.

Conteneurisez votre application

Étape 1 : Exécuter la commande pour créer les fichiers Docker

- Choisissez l'une de vos applications que vous souhaitez conteneuriser et exécutez les commandes suivantes dans un terminal (en remplaçant le chemin) :
 - `cd /chemin/vers/votre/projet/`
 - `docker init`

Conteneurisez votre application

Étape 2 : Suivez les instructions à l'écran

- `docker init` vous guidera à travers quelques questions pour configurer votre projet avec des valeurs par défaut.
- Répondez et appuyez sur Entrée.

Conteneurisez votre application

Étape 3 : Essayez de lancer votre application

- Une fois toutes les questions traitées, exécutez les commandes suivantes dans un terminal pour lancer votre projet (en remplaçant le chemin) :
 - `cd /chemin/vers/votre/projet/`
 - `docker compose up`

Conteneurisez votre application

Étape 4 : Mettre à jour les fichiers Docker si nécessaire

- La commande `docker init` essaie de faire le plus gros du travail, mais des ajustements peuvent être nécessaires.
- Dans ce cas, vous pouvez vous référer aux documentations :
 - Référence Dockerfile:
<https://docs.docker.com/engine/reference/buildr/>

Conteneurisez votre application

Étape 4 : Mettre à jour les fichiers Docker si nécessaire

- Référence des fichiers Compose:

<https://docs.docker.com/compose/compose-file/>

Conteneurisez votre application

Résumé

- Dans ce chapitre, vous avez appris à conteneuriser votre propre application.

Publiez votre image

- Ce guide vous explique comment publier et partager vos images sur **Docker Hub**.

Publiez votre image

Étape 1 : Obtenir l'image d'exemple

- 1. Dans Docker Desktop, cliquez sur la barre de recherche.
- 2. Saisissez `docker/welcome-to-docker`.
- 3. Sélectionnez "Pull" pour télécharger l'image depuis Docker Hub.

Publiez votre image

Étape 2 : Connectez-vous à Docker

- Cliquez sur "Sign in" (en haut à droite de Docker Desktop) pour vous connecter ou créer un nouveau compte Docker.

Publiez votre image

Étape 3 : Renommez votre image

- Avant de publier, vous devez renommer votre image pour que Docker Hub sache qu'elle vous appartient.
- Exécutez la commande suivante dans un terminal (remplacez VOTRE-NOM-UTILISATEUR par votre identifiant Docker) :
 - `docker tag docker/welcome-to-docker VOTRE-NOM-UTILISATEUR/welcome-to-docker`

Publiez votre image

Étape 4 : Transférez votre image vers Docker Hub

- 1. Dans Docker Desktop, allez à l'onglet "Images".
- 2. Dans la colonne "Actions" de votre image, cliquez sur l'icône "Actions".
- 3. Sélectionnez "Push to Hub" (Transférer vers le Hub).

Publiez votre image

Étape 4 : Transférez votre image vers Docker Hub

- Rendez-vous sur Docker Hub et vérifiez que l'image `VOTRE-NOM-UTILISATEUR/welcome-to-docker` apparaît bien dans votre liste de dépôts (repositories).

Publiez votre image

Résumé

- Dans ce guide, vous avez publié et partagé une image sur Docker Hub.

Quiz

- 1. Quel outil est utilisé pour démarrer plusieurs conteneurs avec une seule commande ?
 - A. docker run
 - B. docker build
 - C. docker compose
 - D. docker pull

Quiz

- 2. Qu'est-ce qu'un fichier compose.yaml ?
 - A. Un fichier qui stocke les données d'une application
 - B. Un fichier qui configure l'exécution d'une application multi-conteneur
 - C. Un fichier qui définit l'image d'un conteneur
 - D. Un fichier qui contient les logs d'un conteneur

Quiz

- 3. Quelle est la commande pour supprimer une pile d'applications dans Docker Desktop ?
 - A. docker stop
 - B. docker rm
 - C. docker pull
 - D. Cliquer sur l'icône de suppression dans l'onglet "Containers"

Quiz

- 4. Qu'est-ce qu'un volume Docker ?
 - A. Un emplacement sur votre système de fichiers pour stocker les données d'un conteneur
 - B. Un type de conteneur léger
 - C. Une commande pour exécuter un conteneur en mode détaché
 - D. Un réseau qui relie plusieurs conteneurs

Quiz

- 5. Quelle est la commande pour monter un dossier local dans un conteneur ?
 - A. docker run
 - B. docker build
 - C. docker compose up
 - D. docker volume create

Quiz

- 6. Quel outil est utilisé pour générer les fichiers nécessaires à la conteneurisation d'une application ?
 - A. docker run
 - B. docker build
 - C. docker compose
 - D. docker init

Quiz

- 7. Quelle est la commande pour afficher les logs d'un conteneur ?
 - A. docker run
 - B. docker build
 - C. docker logs
 - D. docker compose up

Quiz

- 8. Où peut-on trouver des images Docker préconfigurées à exécuter ?
 - A. Dans le terminal
 - B. Dans Visual Studio Code
 - C. Sur Docker Hub
 - D. Dans le dossier de votre projet

Quiz

- 9. Qu'est-ce qu'une image Docker ?
 - A. Un instantané statique d'un environnement d'exécution
 - B. Un type de conteneur léger
 - C. Une commande pour exécuter un conteneur en mode détaché
 - D. Un réseau qui relie plusieurs conteneurs

Quiz

- 10. Qu'est-ce qu'un conteneur Docker ?
 - A. Un instantané statique d'un environnement d'exécution
 - B. Une instance en cours d'exécution d'une image Docker
 - C. Une commande pour exécuter un conteneur en mode détaché
 - D. Un réseau qui relie plusieurs conteneurs