

Glossaire

Glossaire Docker : Les termes essentiels

1. Container

- **Définition** : Une instance d'une **image Docker** en cours d'exécution.
- Il s'agit d'un environnement isolé qui contient tout ce qui est nécessaire pour exécuter une application, comme le code, les bibliothèques et les dépendances.
- Les containers sont éphémères : les données non sauvegardées dans un **volume** ou un stockage persistant seront perdues après leur arrêt ou suppression.
- **Exemple** : Démarrer un container à partir d'une image :

```
docker run -d --name my_container nginx
```

2. Image

- **Définition** : Un modèle immuable (en lecture seule) utilisé pour créer des containers.
- Elle contient tout ce dont une application a besoin pour fonctionner : système d'exploitation, bibliothèques système, outils, code source et dépendances.
- Construite à partir d'un **Dockerfile**, elle peut être poussée sur un registre comme Docker Hub pour être partagée.
- **Exemple** : Construire une image à partir d'un fichier Dockerfile :

```
docker build -t my_image:latest .
```

3. Volume

- **Définition** : Un mécanisme pour stocker des données persistantes.

- Contrairement au système de fichiers interne d'un container (éphémère), les volumes permettent de conserver des données même après la suppression d'un container.
- Ils peuvent être partagés entre plusieurs containers et sont stockés sur l'hôte Docker (généralement dans `/var/lib/docker/volumes`).
- **Exemple** : Créer et attacher un volume à un container :

```
docker volume create my_volume
docker run -d --name my_container -v my_volume:/data nginx
```

4. Dockerfile

- **Définition** : Un fichier texte contenant une série d'instructions utilisées pour construire une image Docker.
- Chaque instruction (par exemple, `FROM`, `RUN`, `COPY`) correspond à une étape pour configurer l'image.
- **Exemple** :

```
Dockerfile
CopierModifier
FROM python:3.9
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

5. Registry

- **Définition** : Un service permettant de stocker, partager et télécharger des images Docker.
- **Types** :
 - **Docker Hub** : Registry public par défaut.
 - **Registres privés** : Hébergés sur AWS, GCP ou des serveurs dédiés.

- **Exemple** : Pousser une image sur Docker Hub :

```
docker push myusername/myimage:latest
```

6. Network

- **Définition** : Un mécanisme pour permettre aux containers de communiquer entre eux ou avec l'extérieur.
- **Types de réseaux** :
 - **Bridge** : Réseau isolé par défaut.
 - **Host** : Le container partage l'interface réseau de l'hôte.
 - **Overlay** : Communication entre containers sur différents hôtes.
- **Exemple** : Créer un réseau personnalisé :

```
docker network create my_network
```

7. Bind Mount

- **Définition** : Permet de monter un dossier ou fichier de l'hôte directement dans un container. Contrairement aux volumes, il utilise un chemin absolu spécifique sur l'hôte.
- **Exemple** :

```
docker run -v /path/on/host:/path/in/container my_image
```

8. Compose

- **Définition** : Un outil pour définir et gérer des applications multi-container via un fichier `docker-compose.yml`.
- **Exemple** : Exemple de fichier `docker-compose.yml` : Commande pour démarrer les services :

```
version: '3.8'
services:
```

```
web:
  image: nginx
  ports:
    - "8080:80"
db:
  image: postgres
  environment:
    POSTGRES_PASSWORD: example
```

```
docker-compose up
```

9. Docker Swarm

- **Définition** : Une fonctionnalité native pour orchestrer des containers sur plusieurs hôtes.
- **Exemple** : Initialiser un Swarm :

```
docker swarm init
```

10. Service

- **Définition** : Une abstraction dans Docker Swarm qui déploie une application (un ou plusieurs containers) dans un cluster.
- **Exemple** :

```
docker service create --name webserver -p 80:80 nginx
```

11. Overlay Network

- **Définition** : Un réseau utilisé pour connecter des containers sur différents hôtes dans un cluster Swarm ou Kubernetes.
- **Exemple** :

```
docker network create --driver overlay my_overlay_network
```

12. Entrypoint

- **Définition** : Une instruction dans un Dockerfile qui définit un script ou une commande à exécuter au démarrage d'un container.
- **Exemple** :

```
ENTRYPOINT ["python", "app.py"]
```

13. Tag

- **Définition** : Une version d'une image Docker, utilisée pour identifier différentes versions.
- **Exemple** :

```
docker tag my_image:latest myusername/myimage:v1.0
```

14. Docker Context

- **Définition** : Permet de configurer et de gérer des Docker Hosts distants.
- **Exemple** : Configurer un contexte distant :

```
docker context create remote-host --docker "host=ssh://user@hostname"
```

15. Docker CLI

- **Définition** : Interface en ligne de commande pour gérer containers, images, volumes, réseaux, etc.
- **Commandes courantes** :
 - Lister les containers :

```
docker ps
```

- Supprimer une image :

```
docker rmi my_image
```

Résumé des concepts clés :

Terme	Définition
Container	Instance d'une image Docker, environnement isolé pour exécuter une application.
Image	Modèle immuable utilisé pour créer des containers.
Volume	Stockage persistant partagé ou non entre containers.
Dockerfile	Script décrivant comment construire une image Docker.
Registry	Emplacement pour stocker et partager des images Docker.
Network	Méthode pour connecter des containers ou des services.
Compose	Outil pour gérer des applications multi-container avec des fichiers YAML.

Besoin d'un point précis ou d'un exemple concret ? Dis-le-moi ! 😊

Glossaire MongoDB et MongoDB Compass : Les termes essentiels

MongoDB

1. Base de données

- **Définition** : Une collection logique de données dans MongoDB. Chaque base de données contient plusieurs collections.
- **Exemple** : Créer une base de données dans le shell MongoDB :

```
use my_database;
```

2. Collection

- **Définition** : Un regroupement de documents dans MongoDB, similaire à une table dans une base de données relationnelle.
- Les collections n'imposent pas de schéma fixe (NoSQL).

- **Exemple** : Créer une collection :

```
db.createCollection("my_collection");
```

3. Document

- **Définition** : L'unité de base de MongoDB, stockée dans une collection. Un document est un objet JSON-like, souvent en format BSON (Binary JSON).
- **Exemple** :

```
{
  "_id": "1",
  "name": "John",
  "age": 30,
  "address": {
    "city": "Paris",
    "country": "France"
  }
}
```

4. `_id`

- **Définition** : Champ unique par document dans une collection, utilisé comme clé primaire.
- Par défaut, MongoDB génère automatiquement un `_id` de type ObjectId si vous n'en fournissez pas.
- **Exemple** :

```
{ "_id": ObjectId("60d5f60f8b3e3a2b7c3c5b94"), "name": "Alice" }
```

5. Query

- **Définition** : Une requête permettant de rechercher, insérer, mettre à jour ou supprimer des documents dans MongoDB.
- **Exemple** : Trouver tous les documents où `age > 25` :

```
db.my_collection.find({ age: { $gt: 25 } });
```

6. Index

- **Définition** : Une structure de données qui améliore la vitesse des requêtes. Par défaut, MongoDB crée un index sur le champ `_id`.
- **Exemple** : Créer un index sur le champ `name` :

```
db.my_collection.createIndex({ name: 1 });
```

7. Replica Set

- **Définition** : Un groupe de serveurs MongoDB contenant les mêmes données pour assurer la haute disponibilité.
- Un Replica Set comprend :
 - Un **Primary** (lecture/écriture).
 - Un ou plusieurs **Secondaries** (réplication pour basculement).
- **Exemple** : Initialiser un Replica Set :

```
rs.initiate();
```

8. Sharding

- **Définition** : Une méthode pour partitionner des données entre plusieurs serveurs afin de gérer des volumes massifs.
- **Exemple** : Activer le sharding pour une collection :

```
sh.enableSharding("my_database");
```

9. Aggregation

- **Définition** : Un framework pour effectuer des opérations complexes sur des documents (filtrage, transformation, regroupement, etc.).

- **Exemple** : Pipeline d'agrégation pour regrouper par âge et compter les utilisateurs :

```
db.my_collection.aggregate([  
  { $group: { _id: "$age", count: { $sum: 1 } } }  
]);
```

10. Mongo Shell

- **Définition** : Une interface en ligne de commande pour interagir avec MongoDB.
- **Exemple** : Lister toutes les bases de données :

```
show dbs;
```

MongoDB Compass

1. MongoDB Compass

- **Définition** : Une interface graphique officielle pour interagir avec MongoDB, permettant de visualiser et de gérer des bases de données et collections.
- **Usage** : Outil idéal pour les débutants et pour des tâches de gestion simples.

2. Connection String

- **Définition** : Une URL utilisée pour se connecter à une base de données MongoDB via Compass ou d'autres outils.
- **Exemple** :

```
mongodb+srv://username:password@cluster.mongodb.net/my_d  
atabase
```

3. Schema Tab

- **Définition** : Fonctionnalité dans Compass qui analyse la structure des documents d'une collection.

- **Usage** : Idéal pour comprendre les données lorsque le schéma n'est pas bien défini.
 - **Exemple** : Accéder au schéma depuis l'onglet "**Schema**" dans une collection.
-

4. Aggregation Tab

- **Définition** : Interface pour construire visuellement des pipelines d'agrégation sans écrire de code.
 - **Usage** : Utile pour les utilisateurs non techniques.
 - **Exemple** : Ajouter des étapes d'agrégation comme `$match` ou `$group`.
-

5. Query Builder

- **Définition** : Une fonctionnalité graphique permettant d'écrire des requêtes pour rechercher des documents.
 - **Exemple** : Utiliser l'interface pour filtrer les documents où `age > 25`.
-

6. Export/Import

- **Définition** : Fonctionnalité pour exporter des documents d'une collection ou importer des données dans MongoDB via Compass.
 - **Exemple** : Exporter des données au format JSON ou CSV.
-

7. Index Management

- **Définition** : Onglet permettant de visualiser et de gérer les index d'une collection via Compass.
 - **Exemple** : Créer un index sur plusieurs champs depuis l'interface graphique.
-

8. Performance Advisor

- **Définition** : Outil intégré pour analyser les performances des requêtes et suggérer des optimisations (par exemple, créer des index).
 - **Exemple** : Vérifier les requêtes lentes et appliquer les suggestions.
-

9. Data Explorer

- **Définition** : Vue dans Compass pour explorer les documents d'une collection. Permet de visualiser les données sous forme arborescente.
- **Exemple** : Naviguer et cliquer sur les champs imbriqués dans un document.

10. Validation Rules

- **Définition** : Fonctionnalité pour définir des règles de validation sur une collection pour imposer des contraintes sur les documents.
- **Exemple** : Ajouter une règle qui force le champ `age` à être un entier :

```
{
  "$jsonSchema": {
    "bsonType": "object",
    "required": ["age"],
    "properties": {
      "age": {
        "bsonType": "int",
        "minimum": 0
      }
    }
  }
}
```

Résumé des concepts clés

Terme	Définition
Base de données	Regroupement logique de collections dans MongoDB.
Collection	Regroupement de documents, similaire à une table.
Document	Unité de base dans MongoDB, stockée en format JSON-like (BSON).
Replica Set	Ensemble de serveurs MongoDB pour la réplication et la haute disponibilité.
Sharding	Méthode de partitionnement des données entre plusieurs serveurs.

Aggregation	Framework pour les opérations avancées comme le regroupement ou la transformation des données.
MongoDB Compass	Interface graphique pour gérer MongoDB de manière intuitive.
Connection String	URL de connexion utilisée pour accéder à MongoDB.
Schema Tab	Analyse visuelle des schémas des documents dans Compass.
Performance Advisor	Outil pour identifier les problèmes de performance et proposer des solutions.