Ces commandes constituent un ensemble de fonctionnalités de base pour gérer des conteneurs Docker. Il existe de nombreuses autres commandes Docker pour des cas d'utilisation plus avancés, mais celles-ci couvrent la plupart des opérations courantes. Vous pouvez obtenir plus d'informations sur chaque commande en utilisant **docker --help** suivi du nom de la commande.

1. **docker run**: Cette commande permet de créer et de lancer un conteneur à partir d'une image Docker.
2. **docker build**: Utilisée pour créer une image Docker à partir d'un fichier Dockerfile.
3. **docker pull**: Télécharge une image Docker à partir d'un référentiel Docker Hub ou d'un autre registre.
4. **docker push**: Permet d'envoyer une image Docker vers un registre, ce qui la rend accessible à d'autres utilisateurs.
5. **docker ps**: Affiche la liste des conteneurs en cours d'exécution.
6. **docker ps -a**: Affiche la liste de tous les conteneurs, y compris ceux qui sont arrêtés.
7. **docker images**: Affiche la liste des images Docker locales stockées sur votre système.
8. **docker rmi**: Supprime une image Docker.
9. **docker rm**: Supprime un conteneur.
10. **docker start**: Démarre un conteneur qui a été arrêté.
11. **docker stop**: Arrête un conteneur en cours d'exécution.
12. **docker restart**: Redémarre un conteneur.
13. **docker exec**: Exécute une commande dans un conteneur en cours d'exécution.
14. **docker logs**: Affiche les journaux (logs) d'un conteneur.
15. **docker inspect**: Affiche des informations détaillées sur un conteneur ou une image.
16. **docker network**: Gère les réseaux Docker, y compris la création de réseaux personnalisés.
17. **docker volume**: Gère les volumes Docker pour le stockage de données persistantes.
18. **docker-compose**: Permet de gérer des applications multi-conteneurs à l'aide d'un fichier de configuration **docker-compose.yml**.
19. **docker system prune**: Nettoie le système en supprimant les conteneurs, les images, les réseaux et les volumes inutilisés.
20. **docker info**: Affiche des informations système sur Docker.
21. **docker version**: Affiche la version de Docker installée sur votre système.
22. **docker login**: Permet de s'authentifier le Docker Hub.
23. **docker logout**: Déconnecte du Docker Hub.
24. **docker cp**: Copie des fichiers entre le système hôte et un conteneur.
25. **docker commit**: Crée une nouvelle image à partir des modifications apportées à un conteneur.
26. **docker tag**: Associe un tag à une image, ce qui facilite le partage et la gestion des images.

Ces commandes constituent un ensemble de fonctionnalités de base pour gérer des conteneurs Docker. Il existe de nombreuses autres commandes Docker pour des cas d'utilisation plus avancés, mais celles-ci couvrent la plupart des opérations courantes. Vous pouvez obtenir plus d'informations sur chaque commande en utilisant **docker --help** suivi du nom de la commande.