

■ Docker Cheatsheet (Jour1 → Jour4)

■ *Vérification & Informations*

```
docker --version # Version de Docker
docker info # Infos détaillées sur Docker
```

■ *Images Docker*

```
docker images # Lister les images locales
docker search redis # Rechercher une image sur Docker Hub
docker pull mysql:8.0 # Télécharger une image
docker rmi # Supprimer une image
```

■ *Conteneurs : Cycle de vie*

```
docker ps # Lister les conteneurs actifs
docker ps -a # Inclure les arrêtés
docker run -d -p 8000:80 --name mon-apache httpd # Lancer un conteneur
docker stop # Arrêter un conteneur
docker start # Redémarrer un conteneur
docker rm # Supprimer un conteneur
docker container prune # Supprimer tous les arrêtés
```

■ *Inspection & Logs*

```
docker logs mon-nginx # Journaux d'un conteneur
docker logs -f mon-nginx # Suivi en direct
docker top mon-nginx # Processus actifs
docker inspect mon-nginx # Détails complets
docker stats # Statistiques ressources
```

■ *Conteneurs Interactifs*

```
docker run -it --name ubuntu-test ubuntu /bin/bash # Nouveau conteneur interactif
docker exec -it mon-nginx /bin/bash # Entrer dans un conteneur déjà lancé
```

■ *Gestion avancée*

```
docker rename vieux-nom nouveau-nom # Renommer
docker commit nouvelle-image # Créer image depuis conteneur
```

■ *Réseaux*

```
docker network ls # Lister les réseaux
docker network create mon-reseau # Créer un réseau
docker run -d --name redis-db --network mon-reseau redis
docker run -d --name app-web --network mon-reseau nginx
docker exec -it app-web ping redis-db # Vérifier connectivité
```

■ Volumes & Mounts

```
docker volume create mon-volume
docker volume ls
docker volume inspect mon-volume
docker run -d --name nginx-vol -v mon-volume:/usr/share/nginx/html -p 8082:80
nginx
docker run -d --name nginx-bind -v $(pwd)/site-html:/usr/share/nginx/html -p
8086:80 nginx
```

■ Stratégies de redémarrage

```
docker run -d --name nginx-always --restart always -p 8083:80 nginx
docker run -d --name nginx-fail --restart on-failure:5 -p 8084:80 nginx
```

■ Dockerfile

```
FROM nginx:latest
COPY ./site-html /usr/share/nginx/html
EXPOSE 80
docker build -t mon-nginx .
docker run -d -p 8085:80 --name nginx-custom mon-nginx
```

■ Variables d'environnement

```
docker run -d --name app-env -e APP_ENV=prod -e APP_DEBUG=false nginx
docker run -d --name mysql-env --env-file .env mysql:8.0
```

■ Docker Compose

```
version: '3.8'
services:
  web:
    build: .
    ports:
      - '8088:80'
    depends_on:
      - db
  db:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: appdb
  docker-compose up -d # Lancer
  docker-compose down # Arrêter & supprimer
  docker-compose logs web # Voir logs
```

■ Nettoyage & Optimisation

```
docker system prune -a # Tout nettoyer
docker system df # Espace disque utilisé
```