

# Lab1:

## Lancer un terminal interactif:

```
docker run -it ubuntu bash
```

## Exercices dans le conteneur:

1. Mettre à jour les packages  
    apt update
2. Installer des outils  
    apt install -y curl wget vim
3. Créer des fichiers  
    echo "Hello from container" > /tmp/test.txt  
    cat /tmp/test.txt
4. Quitter le conteneur  
    exit

## Comprendre l'éphémérité:

1. Lancer un nouveau conteneur Ubuntu  
    docker run -it ubuntu bash

2. À l'intérieur, vérifier:

```
ls /tmp/
```

Le fichier test.txt n'existe pas !

C'est un NOUVEAU conteneur

- Concept clé : Éphémérité
- Chaque docker run crée un NOUVEAU conteneur
- Les modifications sont perdues à la sortie
- Les conteneurs sont immuables par design

## Créer sans démarrer

```
docker create --name mon-conteneur nginx
```

## Démarrer

```
docker start mon-conteneur
```

## Voir tous les conteneurs

- Conteneurs en cours d'exécution  
    docker ps
- Tous les conteneurs (incluant arrêtés)  
    docker ps -a

## Arrêter (SIGTERM, puis SIGKILL après 10s)

```
docker stop mon-conteneur
```

## Redémarrer

```
docker restart mon-conteneur
```

**Pause** (freeze les processus)

docker pause mon-conteneur

**Reprendre**

docker unpause mon-conteneur

**Arrêt immédiat** (SIGKILL)

docker kill mon-conteneur

**Supprimer** (doit être arrêté)

docker rm mon-conteneur

**Forcer** la suppression (même en cours)

docker rm -f mon-conteneur

Questions de réflexion

1. Pourquoi les données persistent dans un conteneur arrêté mais pas supprimé?
2. Comment pourrait-on sauvegarder notre application ?
3. Quels sont les avantages de cette éphémérité ?

Réponses

1. Le conteneur arrêté conserve sa couche d'écriture
2. Avec des volumes Docker ou en créant une nouvelle image
3. Environnements propres, reproductibles, pas de "pollution"