

# UNIVERSITE DE KINSHASA



MENTION DE MATHEMATIQUES, STATISTIQUE ET INFORMATIQUE  
B.P 190 KINSHASA XI

## PROJET DU SYSTÈME D'EXPLOITATION L2 LMD

Sujet : Déploiement d'un environnement de conteneurs  
avec Docker et Docker Compose.

Groupe N° : 35



N°	Nom complet
1	TATI NZUZI OBED
2	WONDA KELA GRACIA
3	MOLAMBA MBENGA TRESOR
4	MUKWA MAYELE HERVE
5	GAMA LEMPIA ROMEO
6	BAKISA TAMPAKA TANGO
7	NGAYA NGALAMULUME RACHEL

Prof. Dr Kasengedia Motumbe Pierre

Collaborateurs :

- Doctorant Junior Kanningini
- Fipa Bukusu & Ferdinand Djungu & All

ANNÉE ACADEMIQUE 2025-2026

# Résumé

Ce projet présente la mise en place d'un environnement applicatif basé sur la conteneurisation à l'aide de Docker et Docker Compose.

L'objectif principal est de déployer une application multi-services de manière simple, portable et reproductible.

Grâce aux conteneurs, chaque service fonctionne dans un environnement isolé, ce qui réduit les conflits de dépendances et facilite la maintenance.

L'architecture conçue est composée de plusieurs services, notamment une application web et une base de données.

Docker Compose a été utilisé pour orchestrer ces services, gérer leur communication et automatiser leur démarrage.

Ce travail démontre que la conteneurisation constitue aujourd'hui une solution essentielle pour le déploiement moderne des applications, aussi bien en environnement de développement que de production.

# Abstract

This project focuses on deploying a containerized environment using Docker and Docker Compose. The main objective is to build a multi-service application that is portable, reliable, and easy to manage. Containerization allows each service to run in an isolated environment, reducing dependency conflicts and improving system stability.

The system architecture includes a web application service and a database service.

Docker Compose was used to configure, manage, and run these services together while ensuring proper communication between them.

This project highlights the importance of container technologies as a key solution for modern software deployment and infrastructure management.

# Introduction

Avec l'évolution rapide des technologies informatiques, le déploiement des applications est devenu un défi majeur.

Les différences entre les environnements de développement, de test et de production provoquent souvent des erreurs d'installation et des incompatibilités logicielles.

**Docker** est une technologie de conteneurisation qui permet d'exécuter une application avec toutes ses dépendances dans un conteneur léger et portable.

**Docker Compose**, quant à lui, permet de gérer facilement des applications composées de plusieurs services.

Dans ce projet, nous utilisons Docker et Docker Compose afin de créer un environnement stable, reproductible et simple à déployer.

# Problématique et Objectif

## Problématique

Les méthodes traditionnelles de déploiement des applications présentent plusieurs difficultés :

- Incompatibilités entre systèmes
- Installation complexe des dépendances
- Manque de portabilité
- Difficulté de maintenance

## Objectif principal

Mettre en place un environnement applicatif conteneurisé capable de fonctionner de manière identique sur différentes machines.

## Objectifs spécifiques

- Comprendre le fonctionnement de Docker
- Créer des images Docker personnalisées
- Orchestrer plusieurs services avec Docker Compose
- Tester le bon fonctionnement de l'environnement

# Méthodologie

La réalisation du projet s'est faite en suivant les étapes suivantes:

1. Analyse des besoins et identification des services nécessaires
2. Conception de l'architecture du système
3. Création des fichiers Dockerfile pour les images
4. Configuration du fichier docker-compose.yml
5. Déploiement des conteneurs
6. Tests et validation du fonctionnement global

Cette méthodologie progressive a permis une meilleure organisation du travail.

# Conception du système

L'architecture du système repose sur une approche multi-conteneurs :

- **Service Application Web**

Contient le serveur web et le code de l'application.

- **Service Base de données**

Assure le stockage et la gestion des données.

Les services communiquent via un réseau Docker interne, garantissant sécurité et performance.

# Implémentation ou Configuration

## Dockerfile

Le Dockerfile permet de créer une image personnalisée contenant:

- Le système de base
- Les dépendances nécessaires
- Le code source de l'application

## docker-compose.yml

Le fichier docker-compose.yml définit :

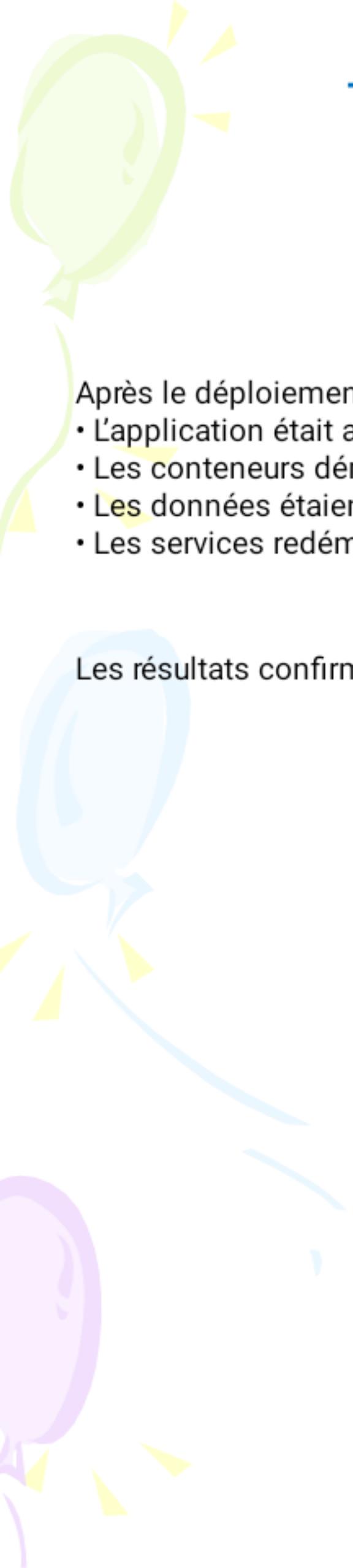
- Les services à lancer
- Les ports exposés
- Les volumes pour la persistance des données
- Les variables d'environnement

## Commandes utilisées

Docker build -t monapp .

Docker-compose up -d

docker ps



## Tests et Résultats

Après le déploiement :

- L'application était accessible via le navigateur
- Les conteneurs démarraient correctement
- Les données étaient conservées grâce aux volumes
- Les services redémarraient sans perte de configuration

Les résultats confirment que l'environnement est fonctionnel et stable.

# Analyse Critique

## Points forts

- Portabilité élevée
- Installation rapide
- Isolation des services
- Facilité de maintenance

## Limites

- Consommation de ressources sur machines peu puissantes
- Complexité initiale pour les débutants
- Configuration avancée requise pour la production à grande échelle

# Conclusion

Ce projet nous a permis de comprendre l'importance de la conteneurisation dans le déploiement des applications modernes. **Docker** et **Docker Compose** offrent une solution efficace pour gérer des environnements complexes de manière simple et fiable.

**La conteneurisation** améliore la portabilité, réduit les erreurs et facilite le travail collaboratif.

Elle représente une compétence essentielle pour les informaticiens d'aujourd'hui.



# Bibliographie

- Documentation officielle Docker, <https://docs.docker.com>  
Documentation Docker Compose,  
<https://docs.docker.com/compose>  
Nigel Poulton, Docker Deep Dive  
Articles techniques sur la conteneurisation (Medium, Dev.to)  
Tutoriels vidéo sur Docker et DevOps