**PROJET FINAL DEVOPS**

* **MEMBRES DU GROUPE** 10 :
  + PANGE FOTSO
  + EDIMA EVINA
  + ANGOUE STEVE DARRYL
  + ABOA NNENGUE MARIE COLETTE
  + ANYENG GHISLAIN JOSUE
  + KENNE ALEX JORDAN
  + TOKAM KOUANKAM
  + KAMENI WIELFRIED

*SOUS LA SUPERVISION DE M. JIHGON BONGI*

ANNEE ACADEMIQUE 2024 - 2025

**SOMMAIRE**

1. *INTRODUCTION GENERALE*

*1.1 Odoo*

*1.2 PgAdmin*

*1.2 Site vitrine*

1. *CONTENEURISATION DE L’APPLICATION*

*2.1 Dockerfile*

*2.2 Tâches*

1. *MISE EN PLACE D’UN PIPELINE CI/CD A L’AIDE DE JENKINS ET ANSIBLE...*

*3.1 Pipeline*

*3.2 Infrastructure*

*3.3 Automatisation du Déploiement*

*3.4 Mise en place du pipeline*

1. *DEPLOIEMENT DES APPLICATIONS DANS UN CLUSTER KUBERNETES*

*Architecture*

*4.1 Déploiement de l’application*

*4.2 Déploiement de PgAdmin*

*4.3 Manifests Kubernetes*

* 1. *Tests et rapport*

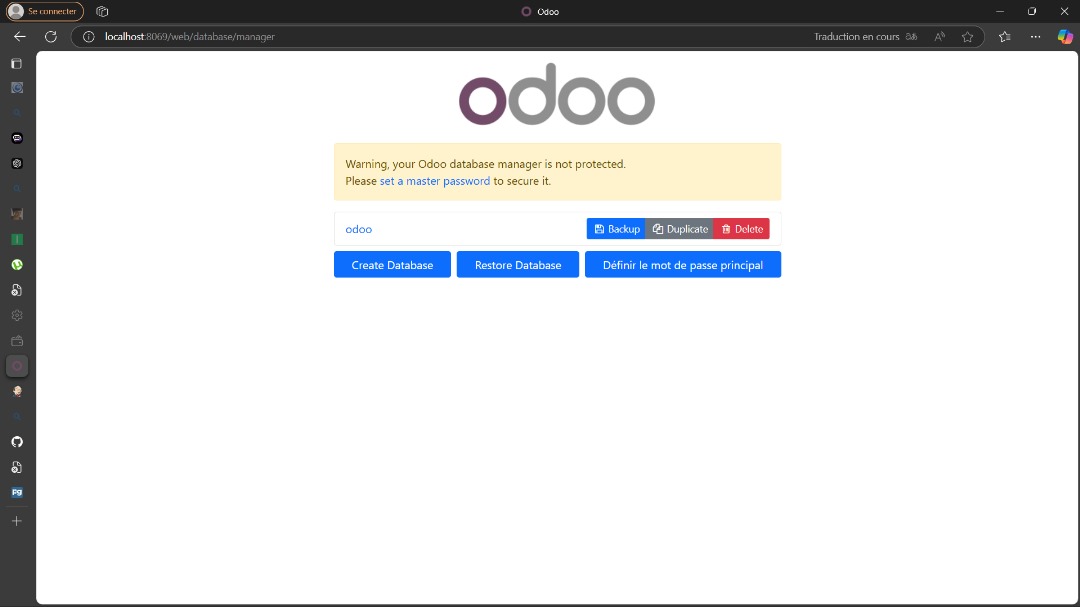
1. *CONCLUSION*
2. **INTRODUCTION**

La société **IC GROUP**, dans laquelle nous travaillions en tant qu’ingénieur DevOps, souhaite mettre sur pied un site web vitrine permettant d’accéder à ses deux applications principales :

1. **Odoo**
2. **pgAdmin**
   1. **Odoo**

Odoo est un ERP multi-usage permettant de gérer les ventes, les achats, la comptabilité, l’inventaire, le personnel, etc. IC GROUP a opté pour la version communautaire d’Odoo (13.0), qui intègre un système LMS (Learning Management System) utilisé pour diffuser des formations internes.

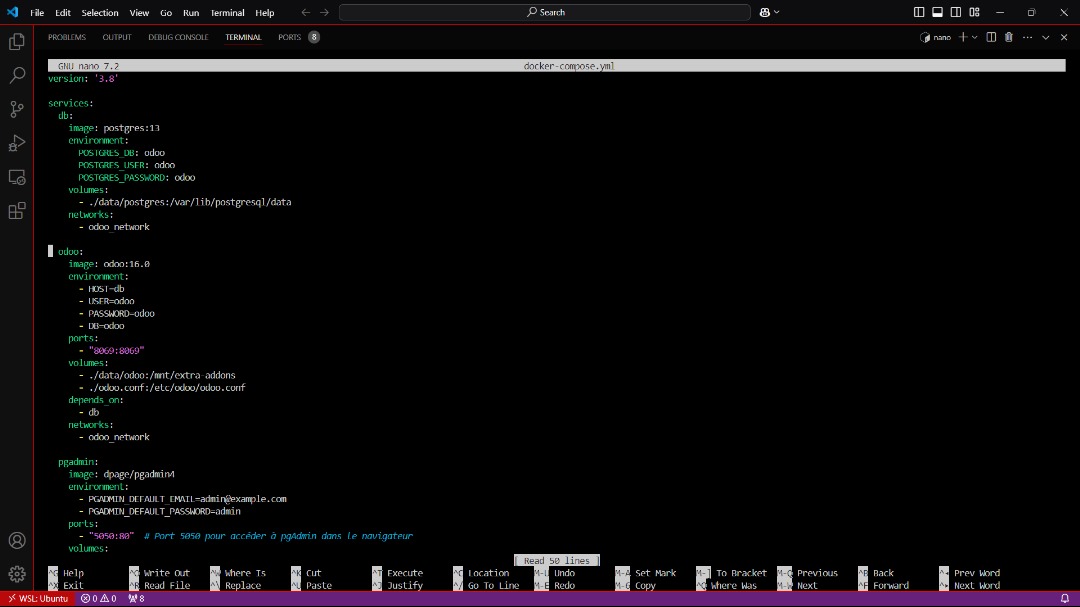
* Site officiel : <https://www.odoo.com/>
* GitHub officiel : <https://github.com/odoo/odoo.git>
* Docker Hub officiel : <https://hub.docker.com/_/odoo>



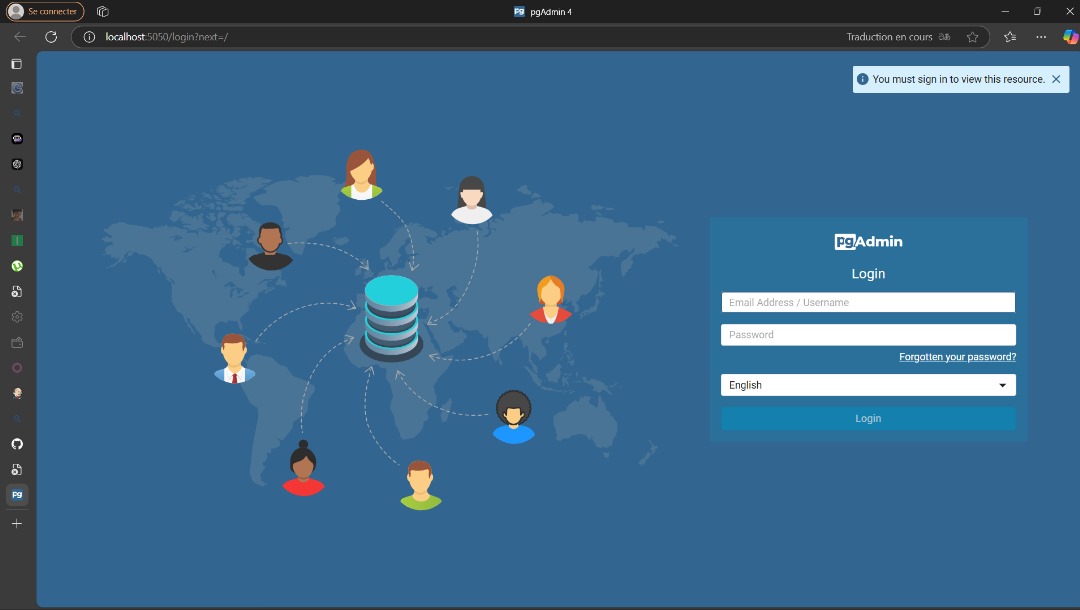
* 1. **PgAdmin**

PgAdmin sera utilisé pour administrer graphiquement la base de données PostgreSQL.

* Site officiel : <https://www.pgadmin.org/>
* Docker Hub officiel : <https://hub.docker.com/r/dpage/pgadmin4/>



**Script du fichier Dockercompose.yml**



**Interface de login PgAdmin**

* 1. **Site Vitrine**

Le site web vitrine a été conçu par l’équipe de développement de notre entreprise. Les fichiers y afférents sont disponibles dans le répertoire suivant : <https://github.com/Legeni07/projet_final.git>. Vous êtes chargé de conteneuriser cette application en permettant la configuration des URL des applications (Odoo et pgAdmin) via des variables d’environnement.

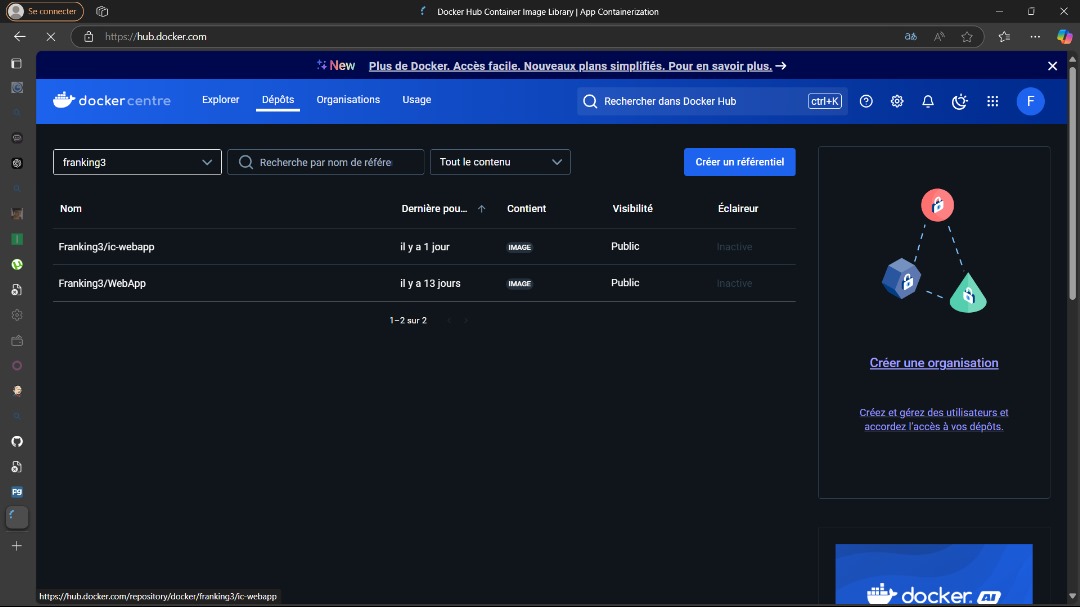
**NB :** L’image créée devra permettre de lancer un conteneur hébergeant ce site web et fournissant les liens vers les applications internes.

1. **Conteneurisation de l’application Web**

L’application web est écrite en Python et utilise le module Flask. Voici les étapes pour sa conteneurisation :

1. **Image de base :** python:3.6-alpine
2. **Répertoire de travail :** /opt
3. **Installation de Flask :** pip install Flask
4. **Port exposé :** 8080
5. **Variables d’environnement :** ODOO\_URL et PGADMIN\_URL
6. **Commande d’exécution :** Lancer app.py via ENTRYPOINT.
   1. **Dockerfile**

* **Nom de l’image :** ic-webapp
* **Tag :** 1.0
* **Nom du conteneur test :** test-ic-webapp



* 1. **Tâche**

1. Créer un Dockerfile suivant les instructions ci-dessus.
2. Build l’image et lancez un conteneur test avec les URL des sites officiels des applications (Odoo et pgAdmin).
3. Supprimer le conteneur test après vérification.
4. Pousser l’image sur votre registre Docker Hub.
5. **Mise en place d’un pipeline CI/CD à l’aide de JENKINS et ANSIBLE**

L’objectif est de mettre en place un pipeline CI/CD pour intégrer et déployer en continu cette solution sur trois serveurs de production

**3.1- Pipeline Stages**

**3.2- Infrastructure**

Les trois serveurs requis sont :

1. **Serveur 1 :** Jenkins (VM ou conteneur)
   * <https://github.com/Legeni07/projet_final/tree/main/jenkins-tools>
2. **Serveur 2 :** Site vitrine + pgAdmin (VM ou conteneur)
3. **Serveur 3 :** Odoo (VM ou conteneur)

3.4- **Automatisation du déploiement**

Nous avons utilisé Docker Compose et Ansible pour créer des rôles facilitant le déploiement des applications.

**Instructions :**

1. Créer un fichier Docker Compose pour :
   * **Odoo** : Créer un réseau et un volume pour persister les données.
   * **pgAdmin** : Configurer la persistance des données et un fichier servers.json personnalisé.
2. Créer deux rôles Ansible :
   * **odoo\_role**
   * **pgadmin\_role**
3. Viabiliser les éléments suivants :
   * Nom du réseau et du volume
   * Répertoire de montage
   * Noms des services et des conteneurs

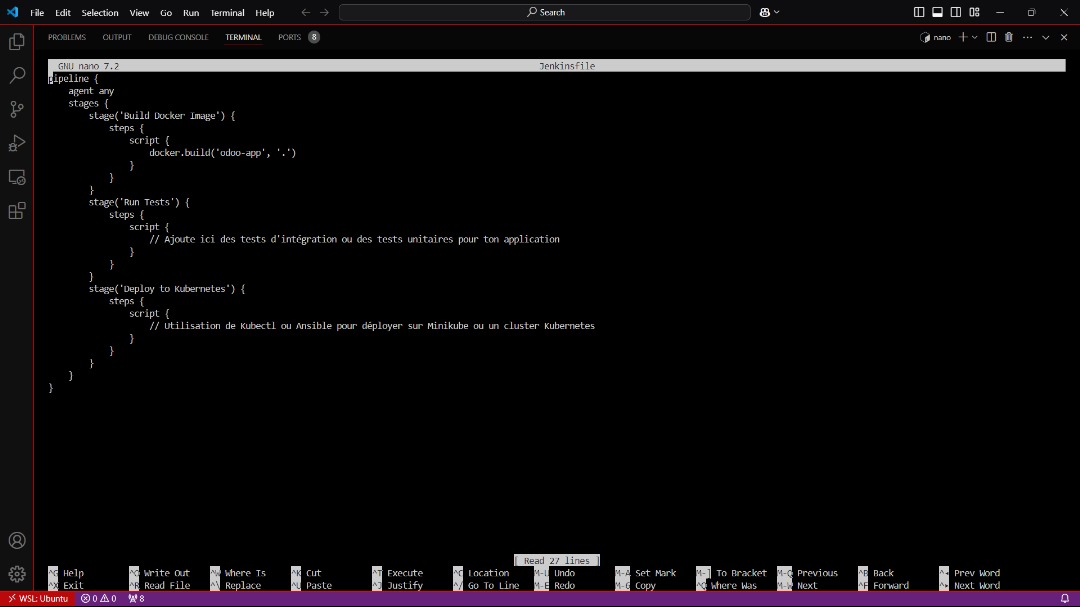
Appeler ces rôles dans le pipeline Jenkins

**3.4- Mise en place du pipeline**

1. Ajouter à la racine du répertoire un fichier releases.txt contenant :
2. ODOO\_URL PGADMIN\_URL Version

https://odoo.com https://pgadmin.org 1.0

1. Modifier le Dockerfile pour :
   * Extraire les valeurs de releases.txt avec awk et export.
2. Créer un Jenkinsfile :
   * Builder l’image avec le tag approprié.
   * Tester les applications.
   * Déployer en production. Tester le pipeline manuellement, puis automatiquement après modification de la version dans releases.txt.



1. **Déploiement des applications dans un cluster Kubernetes**

**4.1- Architecture**

Les applications seront déployées dans un cluster Minikube à un seul nœud suivant l’architecture suivante :

**Tâche**

Identification des ressources (A à H) en précisant leur type et leur rôle.

**4.2- Déploiement de l’application Odoo**

1. Utilisation de l’image Docker appropriée.
2. Assurons-nous que les données sont persistantes dans un répertoire spécifié.

**4.3- Déploiement de pgAdmin**

1. Configuration des données.
2. Fournissons un fichier servers.json préconfiguré pour connecter pgAdmin à la base Odoo.

**4.4- Manifests Kubernetes**

1. Création des manifests pour :
   * Déployer les applications avec les bons réplicas.
   * Assurer la persistance des données.
2. Utilisation de namespace icgroup avec le label env=prod.
3. Protéger les informations sensibles (ex. secrets).

**4.5- Tests et rapport**

1. Lancement les manifests pour déployer les services.
2. Test des applications pour assurer leur bon fonctionnement.



1. **CONCLUSION**

En conclusion, le projet DevOps a permis de transformer de manière significative notre approche du développement et de l'exploitation des applications. Nous avons réduit considérablement les délais de mise en production tout en améliorant la qualité du code. Les résultats obtenus montrent une amélioration notable de la collaboration entre les membres de notre groupe de développement et d'exploitation, ce qui a renforcé notre capacité à livrer des fonctionnalités de manière plus rapide et fiable. Cependant, ce projet a également mis en lumière l'importance de la formation continue et de l'adaptation aux évolutions technologiques, afin de maintenir la pérennité et l'efficacité de la solution DevOps.