



## **UE : DevOps**

### **GROUPE 2**

## **Thème : Projet Final**

#### **MEMBRES DU GROUPE**

- FOPI TCHOUPOU ARNOLD
- GUIA NDO RHODE
- ILOUGAR DESIRER
- KANG MFEGUE FABRICE
- MVOGO BEN ABONG RALPH  
EDOUARD
- ZOUA CHARLOTTE MICHELLE  
ALEXANDRA

**Sous la supervision de : M. BOGNI**

**ANNEE ACADEMIQUE 2024-2025**

## 1. Introduction

### Contexte du Projet

Le projet DevOps a été initié par la société **IC GROUP**, une entreprise spécialisée dans la gestion de systèmes ERP et d'outils de gestion de base de données. Le projet consiste à créer une infrastructure permettant le déploiement continu d'applications via un pipeline CI/CD, tout en intégrant des technologies de conteneurisation et d'orchestration. Ce projet a pour but de déployer une application vitrine pour accéder à deux applications principales utilisées par l'entreprise : **Odoo** et **pgAdmin**.

### Objectifs

Les objectifs du projet sont multiples :

1. **Conteneurisation de l'application web vitrine** utilisant Flask.
2. Mise en place d'un **pipeline CI/CD** utilisant Jenkins et Ansible pour automatiser les processus de construction, de test et de déploiement.
3. **Déploiement des applications dans un cluster Kubernetes** pour garantir la scalabilité et la gestion efficace des ressources.

### Environnement Technologique

Les technologies utilisées pour ce projet incluent :

- **Docker** pour la conteneurisation des applications.
- **Jenkins** pour l'automatisation du pipeline CI/CD.
- **Kubernetes (Minikube)** pour le déploiement et l'orchestration des containers.
- **Ansible** pour l'automatisation de la configuration de l'infrastructure.

## 2. Conteneurisation de l'application web

### Description de l'application

L'application vitrine a été conçue en Python avec le module **Flask**. Elle sert à fournir une interface permettant d'accéder à deux services principaux : **Odoo** (ERP) et **pgAdmin** (outil de gestion des bases de données PostgreSQL). L'application utilise Flask pour gérer les requêtes et afficher les liens vers les applications internes.

### Processus de conteneurisation

1. **Création du Dockerfile :**

- L'image de base utilisée est python:3.6-alpine.
- Le répertoire /opt a été défini comme répertoire de travail.
- Le module Flask a été installé via pip install.
- Le port 8080 a été exposé pour accéder à l'application Flask.
- Les variables d'environnement ODOO\_URL et PGADMIN\_URL ont été créées pour permettre la personnalisation des URL des applications Odoo et pgAdmin.
- L'application a été lancée via ENTRYPOINT en exécutant le fichier app.py.

## 2. Test de l'image Docker :

- L'image Docker a été construite avec succès et un container de test a été lancé pour vérifier que l'application vitrine fonctionne correctement en accédant aux sites officiels de **Odoo** et **pgAdmin**.

## 3. Publication sur Docker Hub :

- Après les tests, l'image Docker a été poussée sur le registre Docker Hub pour pouvoir être utilisée dans d'autres environnements.

## 3. Mise en place du pipeline CI/CD

### Objectif du pipeline

L'objectif du pipeline CI/CD est d'automatiser les étapes de construction, de test et de déploiement de l'application vitrine, en garantissant une livraison continue et sans erreur des nouvelles versions. Le pipeline a été conçu pour fonctionner avec **Jenkins** et **Ansible**, permettant ainsi une gestion fluide des processus de déploiement sur des serveurs de production.

### Stages du pipeline

#### 1. Stage 1 : Construction de l'image Docker :

- Le pipeline commence par la construction de l'image Docker à partir du Dockerfile créé dans la première partie du projet.

#### 2. Stage 2 : Tests unitaires :

- Des tests unitaires sont effectués pour valider que l'application vitrine fonctionne correctement avant d'être déployée en production. Ces tests

incluent la validation de la connectivité aux services Odoo et pgAdmin via les URL définies dans les variables d'environnement.

### 3. **Stage 3 : Déploiement en environnement de production :**

- L'image Docker est déployée sur les serveurs de production à l'aide de **Jenkins** et **Ansible**. Des rôles Ansible ont été créés pour automatiser la configuration des environnements et l'exécution des containers Docker.

### 4. **Stage 4 : Validation du déploiement :**

- Une fois le déploiement effectué, des tests de validation sont réalisés pour s'assurer que l'application est accessible et fonctionne correctement en production.

#### **Fichier releases.txt**

Un fichier releases.txt a été créé à la racine du dépôt. Ce fichier contient des informations cruciales concernant les versions des applications et leurs URL, qui sont récupérées automatiquement lors de la construction de l'image Docker à l'aide des commandes awk et export.

## **4. Déploiement sur Kubernetes**

### **Architecture Kubernetes**

L'architecture de déploiement sur **Kubernetes** repose sur un cluster Minikube, qui simule un environnement de production à un seul nœud. Les applications sont déployées à l'aide de **pods**, **services** et **volumes persistants** pour garantir la scalabilité et la persistance des données.

### **Déploiement d'Odoo**

Odoo a été déployé en tant qu'application à 2 tiers dans Kubernetes. Un **Deployment** a été utilisé pour garantir la réplication des instances Odoo et un **Service** de type ClusterIP a été configuré pour accéder à l'application. Les données de la base de données sont persistées à l'aide d'un volume Kubernetes monté sur un répertoire spécifique sur l'hôte.

### **Déploiement de PgAdmin**

PgAdmin a été déployé de manière similaire à Odoo, mais avec un focus particulier sur la persistance des données et la configuration de la connexion automatique à la base de données Odoo. Un fichier de configuration personnalisé pour PgAdmin a été monté en tant que volume dans le container.

### **Tests de déploiement**

Une fois le déploiement effectué sur Kubernetes, des tests ont été réalisés pour vérifier l'accessibilité des applications et leur bon fonctionnement.

## 5. Conclusion

### Résumé des réalisations

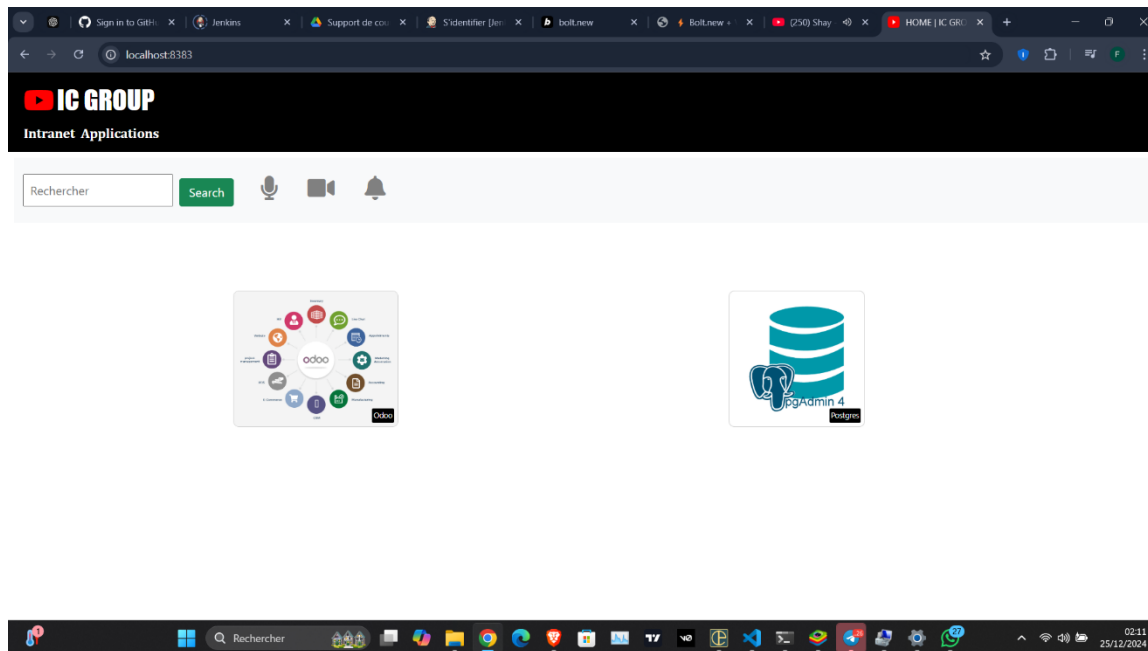
- **Conteneurisation réussie** de l'application vitrine Flask, permettant une gestion flexible des applications Odoo et pgAdmin.
- Mise en place d'un **pipeline CI/CD** fonctionnel avec **Jenkins** et **Ansible**, assurant l'automatisation du processus de déploiement.
- Déploiement des applications dans un **cluster Kubernetes** avec gestion de la persistance des données et configuration des volumes.

### Défis rencontrés

- La configuration des services dans Kubernetes a nécessité une attention particulière pour garantir la persistance des données, notamment pour Odoo et PgAdmin.
- La mise en place du pipeline CI/CD a exigé la gestion des dépendances entre les différentes étapes du pipeline, ainsi que la configuration adéquate des environnements de test et de production.

### Perspectives

- **Amélioration de la scalabilité** : Le déploiement sur Kubernetes permettra d'ajuster les ressources en fonction de la demande.
- **Mise en production continue** : L'intégration du pipeline avec des outils de monitoring et de notification pourrait améliorer encore le processus de déploiement continu.



```
fops@fops: /mnt/d/projet_de x + v
Handling connection for 8081
E1231 05:05:34.175477 57276 portforward.go:413] "Unhandled Error" err=<
  an error occurred forwarding 8081 -> 8069: error forwarding port 8069 to pod 3345594ba387fa688ebd75503ed8204c055109a519fe52c92d97f9ccd64fbdd, uid :
  exit status 1: 2024/12/31 04:05:34 socat[14417] E connect(5, AF=2 127.0.0.1:8069, 16): Connection refused
>
error: lost connection to pod
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl get pods -n icgroups
NAME                                READY   STATUS    RESTARTS   AGE
odoo-7b56d64dfc-2j7jq              0/1     CrashLoopBackOff   6 (2m11s ago)    13m
odoo-db-77c5f7d54c-vvwcw           1/1     Running      0           14m
pgadmin-6797d98489-xrhbh           1/1     Running      0           10m
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl logs odoo-7b56d64dfc-2j7jq -n icgroups
Database connection failure: could not translate host name "db" to address: Temporary failure in name resolution
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl get secret odoo-db-secret -n icgroups -o yaml
apiVersion: v1
data:
  password: YWRtaW4xMjM=
kind: Secret
metadata:
  annotations:
    kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"password":"YWRtaW4xMjM="},"kind":"Secret","metadata":{"annotations":{},"name":"odoo-db-secret","namespace":"icgroups"},"type":"Opaque"}
  creationTimestamp: "2024-12-31T03:55:41Z"
  name: odoo-db-secret
  namespace: icgroups
  resourceVersion: "2111"
  uid: ef58880c-c3f8-4b79-bf67-d48362b3521e
type: Opaque
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl delete pod odoo-7b56d64dfc-2j7jq -n icgroups
pod "odoo-7b56d64dfc-2j7jq" deleted
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl get pods -n icgroups
NAME                                READY   STATUS    RESTARTS   AGE
odoo-7b56d64dfc-dgpgs              1/1     Running    0           30s
odoo-db-77c5f7d54c-vvwcw           1/1     Running    0           16m
pgadmin-6797d98489-xrhbh           1/1     Running    0           12m
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl port-forward svc/odoo-service 8081:86 -n icgroups
Forwarding from 127.0.0.1:8081 -> 8069
Forwarding from [::1]:8081 -> 8069

17°C
Partly cloudy
Rechercher
05:09
31/12/2024

fops@fops: /mnt/d/projet_de x + v
... // 17 identical fields
}

fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-pv.yaml
persistentvolume/odoo-pvs created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-pvc.yaml
persistentvolumeclaim/odoo-pvc created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-deployment.yaml
deployment.apps/odoo created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-service.yaml
service/odoo-service created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-db-secret.yaml
Error from server (BadRequest): error when creating "odoo-db-secret.yaml": Secret in version "v1" cannot be handled as a Secret: illegal base64 data at input byte 0
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f odoo-db-secret.yaml
secret/odoo-db-secret created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f pgadmin-deployment.yaml
deployment.apps/pgadmin created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl apply -f pgadmin-service.yaml
service/pgadmin-service created
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl get pods -n icgroup
NAME                                READY   STATUS    RESTARTS   AGE
odoo-5dffd5865-4qd4d              1/1     Running    0           30m
odoo-5dffd5865-dmg68              1/1     Running    0           30m
pgadmin-558867fbff-tw6cz          1/1     Running    0           29m
postgres-69fd7fc867-mwsjw         1/1     Running    0           30m
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl get pods -n icgroups
NAME                                READY   STATUS    RESTARTS   AGE
odoo-7b56d64dfc-2j7jq              1/1     Running    2 (29s ago)    3m16s
odoo-db-77c5f7d54c-vvwcw           1/1     Running    0           4m10s
pgadmin-6797d98489-xrhbh           1/1     Running    0           42s
fops@fops: /mnt/d/projet_devops/projet_final$ kubectl port-forward svc/pgadmin-service 5050:80 -n icgroup
Forwarding from 127.0.0.1:5050 -> 80
Forwarding from [::1]:5050 -> 80
Handling connection for 5050
Handling connection for 5050
Handling connection for 5050
Handling connection for 5050
Handling connection for 5050

Gestionnaire des tâches

17°C
Partly cloudy
Rechercher
04:59
31/12/2024
```

ChatGPT pgAdmin 4 10.98.139.23 127.0.0.1:5050/login?next=/

17°C Partly cloudy 04:58 31/12/2024

localhost:8185/manage/pluginManager/updates/ Tous les favoris

# Jenkins

rechercher (CTRL+K) fops se déconnecter

Tableau de bord > Administrer Jenkins > Plugins

## Plugins

- Mises à jour
- Plugins disponibles
- Plugins installés
- Paramètres avancés
- Progression des téléchargements**

## Progression du téléchargement

Préparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Pipeline: REST API	Succès
Authentication Tokens API	Succès
Docker Commons	Succès
Docker Pipeline	Succès
Loading plugin extensions	Success
Ansible	Succès
Loading plugin extensions	Success
Cloud Statistics	Succès
Apache HttpComponents Client 5.x API	Succès
Commons Compress API	En cours d'installation
Docker API	En cours
Docker	En cours
Loading plugin extensions	Pending

02:31 25/12/2024



hub.docker.com/?\_gl=1\*68ck1\*\_ga\*NDM0MzI3NDk1JjE3MzQzOTQ0Znc\*\_ga\_XWPMQMYHQ\*MTc2NTA0TE4NC41UjEuMTc2NTA0TE40S41NS4wJJA

New More Docker. Easy Access. New Streamlined Plans. Learn more. →

dockerhub Explore Repositories Organizations Usage Search Docker Hub ctrl+K

fops4 Search by repository name All content Create a repository

Name	Last Pushed ↑	Contains	Visibility	Scout
fops4/ic-webapp	less than a minute ago IMAGE		Public	Inactive

1-1 of 1

Create an organization  
Create and manage users and grant access to your repositories.

By clicking "Accept All Cookies," you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts. Cookies Settings Reject All Accept All Cookies

```
fops@fops: /mnt/d/projet_de
efa76becf38b: Preparing
671e3248113c: Waiting
1965cfbef2ab: Waiting
6d3ac3489996: Waiting
denied: requested access to the resource is denied
fops@fops:/mnt/d/projet_devops/projet_final$ docker login

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: BHQT-TSVQ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser..
*Login canceled
fops@fops:/mnt/d/projet_devops/projet_final$
fops@fops:/mnt/d/projet_devops/projet_final$ docker login -u fops4
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
fops@fops:/mnt/d/projet_devops/projet_final$ docker login -u fops4
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
fops@fops:/mnt/d/projet_devops/projet_final$ docker login -u fops4
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
fops@fops:/mnt/d/projet_devops/projet_final$ docker login -u fops4
Password:
WARNING! Your password will be stored unencrypted in /home/fops/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
fops@fops:/mnt/d/projet_devops/projet_final$ docker tag ic-webapp:1.0 fops4/ic-webapp:1.0
fops@fops:/mnt/d/projet_devops/projet_final$ docker push fops4/ic-webapp:1.0
The push refers to repository [docker.io/fops4/ic-webapp]
48354f94654f: Pushed
a8ae0b344d7d: Pushed
5f70bf18a086: Mounted from library/wordpress
3156423bd38f: Mounted from library/python
efa76becf38b: Mounted from library/python
671e3248113c: Pushed
1965cfbef2ab: Pushed
6d3ac3489996: Mounted from library/python
1.0: digest: sha256:085514e29eae49edb612898a71ce4a9a1163eccc4481e35c09fd342d36828e33 size: 1996
fops@fops:/mnt/d/projet_devops/projet_final$
```

Correction\_Algorithmes.pdf | Torrent9.site - Télécharger au... | Torrent9.site - recherche de vi... | bel air - Recherche Google | YourTorrent Software is Up! | Tableau de bord [jenkins] x + -

localhost:8185

Gmail | YouTube | Maps

**Jenkins** | rechercher (CTRL+K) | fops | se déconnecter

Tableau de bord >

+ Nouveau Item

Historique des constructions

Relations entre les builds

Vérifier les empreintes numériques

Administrer Jenkins

Mes vues

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations 0/2

Ajouter une description

Tous +

S	M	Nom du projet	Dernier succès	Dernier échec	Dernière durée
⊗	☁	Pipeline-Deploy-Odoo-PgAdmin	s. o.	1 h 16 mn #23	10 mn

Icônes: S M L

REST API | Jenkins 2.479.2

17°C | Partly cloudy | Rechercher | 05:19 | 31/12/2024

```
fops@fops:/mnt/d/projet_devops/projet_final$ docker run --name test-ic-webapp -e OD00_URL=https://www.odoo.com -e PGADMIN_URL=https://www.pgadmin.org -p 8080:8080 ic-webapp:1.0
docker: Error response from daemon: driver failed programming external connectivity on endpoint test-ic-webapp (d5478e7518963749237afeca8af0a2da1d4116e858987d357f21c6708b9ba5dc): failed to bind port 0.0.0.0:8080/tcp: Error starting userland proxy: listen tcp4 0.0.0.0:8080: bind: address already in use.
fops@fops:/mnt/d/projet_devops/projet_final$ docker run --name test-ic-webapp -e OD00_URL=https://www.odoo.com -e PGADMIN_URL=https://www.pgadmin.org -p 8383:8080 ic-webapp:1.0
docker: Error response from daemon: Conflict. The container name "/test-ic-webapp" is already in use by container "c13bbb337404d0a215a0e6a36c8d4784133c2d581a3e39137fa7ac3d8526fbc6". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
fops@fops:/mnt/d/projet_devops/projet_final$ docker rm test-ic-webapp
test-ic-webapp
fops@fops:/mnt/d/projet_devops/projet_final$ docker run --name test-ic-webapp -e OD00_URL=https://www.odoo.com -e PGADMIN_URL=https://www.pgadmin.org -p 8383:8080 ic-webapp:1.0
This is a sample web application for intranet applications display.

No Command line argument. Odoo url from environment variable =https://www.odoo.com
No Command line argument. Pgadmin url from environment variable =https://www.pgadmin.org
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
  * Running on http://172.17.0.2:8080/ (Press CTRL+C to quit)
172.17.0.1 - - [25/Dec/2024 01:10:57] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Dec/2024 01:10:57] "GET /static/CSS/style.css HTTP/1.1" 200 -
172.17.0.1 - - [25/Dec/2024 01:10:57] "GET /static/CSS/bootstrap.min.css HTTP/1.1" 200 -
172.17.0.1 - - [25/Dec/2024 01:10:57] "GET /static/images/odoo.jpg HTTP/1.1" 200 -
172.17.0.1 - - [25/Dec/2024 01:10:57] "GET /static/images/PgAdmin4.png HTTP/1.1" 200 -
172.17.0.1 - - [25/Dec/2024 01:11:00] "GET /static/images/ing-icon1.png HTTP/1.1" 200 -
```

Rechercher | 02:11 | 25/12/2024