

# PROJET DEVOPS - Orchestration

## I) Introduction

La société **IC GROUP** dans laquelle vous travaillez en tant qu'ingénieur DevOps souhaite mettre sur pied un site web vitrine devant permettre d'accéder à ses 02 applications phares qui sont :

- 1) Odoo et
- 2) pgAdmin

**Odoo**, un ERP multi usage qui permet de gérer les ventes, les achats, la comptabilité, l'inventaire, le personnel ...

Odoo est distribué en version communautaire et Enterprise. IC GROUP souhaite avoir la main sur le code et apporter ses propres modifications et customisations ainsi elle a opté pour l'édition communautaire. Plusieurs versions de Odoo sont disponibles et celle retenue est la 13.0 car elle intègre un système de LMS (Learning Management System) qui sera utiliser pour publier les formations en internes et ainsi diffuser plus facilement l'information.

Liens utiles :

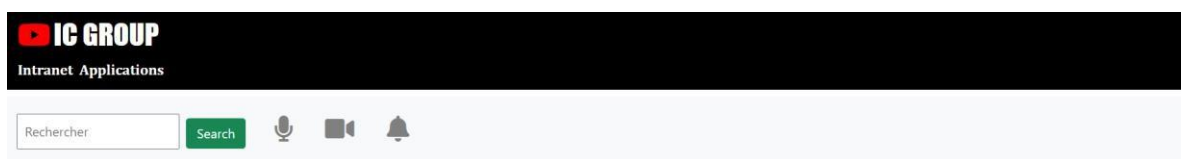
- Site officiel : <https://www.odoo.com/>
- GitHub officiel: <https://github.com/odoo/odoo.git>
- Docker Hub officiel : [https://hub.docker.com/\\_/odoo](https://hub.docker.com/_/odoo)

**pgAdmin** quant à elle devra être utilisée pour administrer de façon graphique la base de données PostgreSQL créée précédemment.

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

Le site web vitrine a été conçu par l'équipe de développeurs de l'entreprise et les fichiers relatifs à cette application web se trouvent dans le repo suscité : <https://github.com/sadofrazer/ic-webapp.git>. Il est de votre responsabilité de conteneuriser cette application tout en permettant la saisie des différentes URL des applications (Odoo et pgadmin) par le biais des variables d'environnement

Ci-dessous un aperçu du site vitrine attendu.



**NB** : L'image créée devra permettre de lancer un container permettant d'héberger ce site web et ayant les liens adéquats permettant d'accéder à nos applications internes

## II) Conteneurisation de l'application web.

Il s'agit en effet d'une application web python utilisant le module Flask. Les étapes à suivre pour la conteneurisation de cette application sont les suivantes :

- 3) Image de base : `python:3.6-alpine`
- 4) Définir le répertoire `/opt` comme répertoire de travail
- 5) Installer le module Flask à l'aide de `pip install`
- 6) Exposer le port `8080` qui est celui utilisé par défaut par l'application
- 7) Créer les variables d'environnement `ODOO_URL` et `PGADMIN_URL` afin de permettre la définition de ces url lors du lancement du container
- 8) Lancer l'application `app.py` dans le `ENTRYPOINT` grâce à la commande `python`

Une fois le Dockerfile créé, Buildez le et lancer un container test permettant d'aller sur les sites web officiels de chacune de ces applications (site web officiels fournis ci-dessus).

**Nom de l'image :** `ic-webapp` ;

**tag:** `1.0`

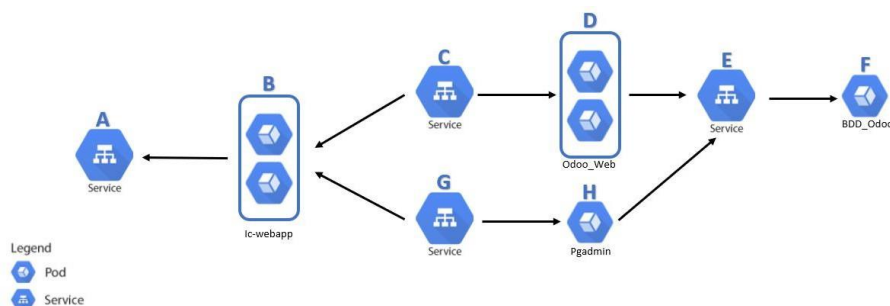
**container test\_name:** `test-ic-webapp`

Une fois le test terminé, supprimez ce container test et poussez votre image sur votre registre Docker hub.

## III) Partie 1 : Déploiement des différentes applications dans un cluster Kubernetes.

### a. Architecture

Les applications ou services seront déployées dans un cluster Minikube, donc à un seul nœud et devront respecter l'architecture suivante.



En vous basant sur cette architecture logicielle, bien vouloir identifier en donnant le type et le rôle de chacune des ressources (A...H) mentionnées dans cette architecture.

### b. Déploiement de l'application Odoo

Comme décrite ci-dessus, Odoo est une application web de type 2 tier contenant différents modules facilitant la gestion administrative d'une société.

En Vous servant des différents liens mentionnés ci-dessus, déployer Odoo à l'aide des images docker correspondantes et assurez vous que les données de la base de données Odoo soit persistantes et sauvegardées dans un répertoire de votre choix sur votre hôte. **NB:** respectez l'architecture ci-dessus

## c. Déploiement PgAdmin

Comme ci-dessus, servez-vous de la documentation de déploiement de PgAdmin sous forme de container afin de déployer votre application.

Vous devez par la suite découvrir dans la documentation, le répertoire contenant les données et paramètres de l'application PgAdmin afin de le rendre persistant.

Notez également que PgAdmin est une application web d'administration des bases de données PostgreSQL, Toutefois, le déploiement d'un container PgAdmin ne nécessite pas obligatoirement la fourniture des paramètres de connexion à une BDD, donc vous pouvez initialement déployer l'interface web en fournissant le minimum de paramètres requis (adresse mail + mot de passe) et ce n'est que par la suite par le biais de l'interface graphique que vous initiez les différentes connexion à vos bases de données.

Afin de réduire le nombre de tâches manuelles, nous souhaiterons qu'au démarrage de votre container PgAdmin, que ce dernier ait automatiquement les données nécessaires lui permettant de se connecter à votre BDD Odoo. Pour ce faire, il existe un fichier de configuration PgAdmin que vous devrez au préalable customiser et fournir par la suite à votre container sous forme de volume.

Ce fichier doit être situé au niveau du container dans le répertoire : `/pgadmin4/servers.json`

```
{
  "Servers": {
    "1": {
      "Name": "Minimally Defined Server",
      "Group": "Server Group 1",
      "Port": 5432,
      "Username": "postgres",
      "Host": "localhost",
      "SSLMode": "prefer",
      "MaintenanceDB": "postgres"
    },
    "2": {
      "Name": "Minimally Defined Server",
      "Group": "Server Group 1",
      "Port": 5432,
      "Username": "postgres",
      "Host": "localhost",
      "SSLMode": "prefer",
      "MaintenanceDB": "postgres"
    }
  }
}
```

## d. Déploiement des différentes applications

En vous servant des données ci-dessus, créez les différents manifests correspondants aux ressources nécessaires au bon fonctionnement de l'application tout en respectant l'architecture fournie (Nbre de réplicas et persistance de données).

Notez également que l'ensemble de ces ressources devront être créés dans un namespace particulier appelé «*icgroup*» et devront obligatoirement avoir toutes au moins le label «*env = prod*»

**NB :** Etant donné que vos manifests pourront être publics (pousser vers un repo Git), bien vouloir prendre les mesures nécessaires afin d'utiliser les ressources adéquates permettant de cacher vos informations sensibles.

## e. Test de fonctionnement et rapport final

Lancez l'exécution de vos différents manifests afin de déployer les différents services ou applications demandés, testez le bon fonctionnement de vos différentes application et n'hésitez pas à prendre des captures d'écran le plus possible afin de consolider votre travail dans un rapport final qui présentera dans les moindre détails ce que vous avez fait.

## IV) ANNEXE

Ci-dessous un exemple de description des qualifications souhaitées pour un poste de Devops

### Qualifications

#### Les technos à maîtriser (tout ou partie) :

- Une expertise OS parmi Linux/Windows
- Bonnes connaissances du cloud Azure et/ou GCP
- Expertises Azure Devops souhaitées
- Compétences clés : CI/CD, gitlab-ci, Jenkins, Kubernetes, helm, docker, kubernetes, istio, azure devops et autres services ci/cd cloud provider.
- Connaissances des méthodes de travail agile.

#### Le profil recherché :

- Profil junior ou confirmé ;
- Bon niveau de Scripting (python) et d'utilisation d'API ;
- Autonome ;
- Tenace ;
- Créatif, dans la recherche de solutions notamment ;
- Capacités rédactionnelles ;
- Pédagogue (vous serez amené à former les personnes qui intégreront l'équipe par la suite) ;
- Anglais courant dans un contexte pro.
- Apte à se déplacer chez le client et ouvert rotation astreinte.

**NB :** Bien vouloir prêter attention aux qualités encadrées en jaune ci-dessus en jaune, vous vous rendez compte en effet que maîtriser les technologies seulement ne suffit pas, il faut en plus de ça avoir un esprit très créatif, de très bonnes capacités rédactionnelles pour rédiger vos différents rapports et également des qualités de pédagogue qui vous aideront à parfaire les explications de vos actions dans vos différents rapports afin de faciliter leur compréhension.

**Compte tenu de tout cela, je vous invite tous à donner l'importance à ce volet « rapport » de votre projet final, car c'est également une partie très importante qui devra pouvoir décrire le contenu de l'ensemble de votre travail.**

Merci de le rédiger correctement avec les captures d'écran, commentaires et explications qui vont bien car cette partie sera prise en compte dans votre note finale.