*TP TAKIMA DOCKER*

*Why should we run the container with a flag -e to give the environment variables?*

Ca permet de passer des paramètres d’environnement comme des variables de connexion d’une base de donnée. Le but est de ne pas mettre en dur les identifiants, mdp de connexion dans le code.

*Why do we need a volume to be attached to our postgres container?*

Un volume permet de garder les données après un redémarrage ou une suppression du conteneur. Ca permet de garantir la sauvegarde des infos de la base de donnée notamment.

* 1. *Document your database container essentials: commands and Dockerfile.*

Pour configurer un conteneur database, on utilise les commandes suivantes :

* Docker build -t nomImage . => permet de générer l’image de la database
* Docker run –name nomConteneur –network= app-network -d -p 5432:5432 nomImage => permet de lancer le conteneur qui se connecte avec les autres conteneur du réseau app-network (back et front) sur les port 5432

Dans le Dockerfile il faut mettre les identifiants de connexion à la base de donnée : db, password, user et url.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

*1-2 Why do we need a multistage build? And explain each step of this dockerfile.*

Un multistage build permet de créer des images plus légère en séparant les images et les dépendances pour permettre une meilleur exécution.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

From => permet de récupérer l’image souhaité

ENV => permet de récupérer les variables d’environnement

MKDIR => Créer le répertoire de travail de l’app

COPY => permet d’overwritter le fichier

RUN => lance l’image dans le conteneur

*Why do we need a reverse proxy?*

Un reverse proxy ets un intermédiaire entre les clients et le serveur back-end. Il peut répartir les requêtes sur plusieurs back-end. Il se place en amont du back-end pour redistribuer les requêtes.

*Why is****docker-compose****so important?*

Docker compose permet de définir et gérer les conteneurs pour l’application 3-tiers à l’aide d’un simple fichier YAML. Dans le fichier YAML, on défini chaque partie de l’app et on peut lancer toute l’app avec un simple commande : *docker compose up ---build.*

*1-3Document docker-compose most important commands. 1-4 Document your docker-compose file.*

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementDans le docker compose, on défini les 3 services : backend, frontend et database.

Dans chaque service, on défini le context qui est le chemin vers le fichier et les ports utilisés par le conteneur.

Environnement permet de définir les variables d’environnements qui seront utiles pour l’app.

Netwroks permet de définir sur quel réseau se situe le conteneur. Ici, tous les conteneurs sont surle même réseaux pour qu’ils puissent *communiquer.*

*Une image contenant texte, capture d’écran, Police

Description générée automatiquement*

*On ajoute aussi le volume dans le docker compose pour sauvegarder les valeurs même après suppression du conteneur.*

*Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement*

*Ici, on défini le network et le volumes*

*1-5 Document your publication commands and published images in dockerhub.*

Il faut tag l’image du docker avec : *docker tag image USERNAME/image :1.0*

Puis on le push, pour qu’il soit accessible sur le docker hub avec : *docker push USERNAME/image :1.0*

*Why do we put our images into an online repo?*

Mettre une image en ligne permet à d’autres utilisateurs de pouvoir y accéder et collaborer. Ca rend le projet scalable sur tous les environnements.