Admin Infra : Labo 5

Préliminaires

Pour cette séance, nous continuons d'utiliser notre Debian sur Azure.

Lorsqu'un mot de passe est demandé, nous vous conseillons de mettre « azerty1.» par facilité.

Exercice 1 – Manipuler docker compose

Nous continuons à construire l'infrastructure de notre société ITDEV. Celle-ci souhaite que ses développeurs complètent leur formation des images docker en utilsant docker compose.

Référez-vous à la section 11.3 du syllabus. Lisez-bien les indications utiles ci-dessous avant de commencer l'exercice.

Indications utiles

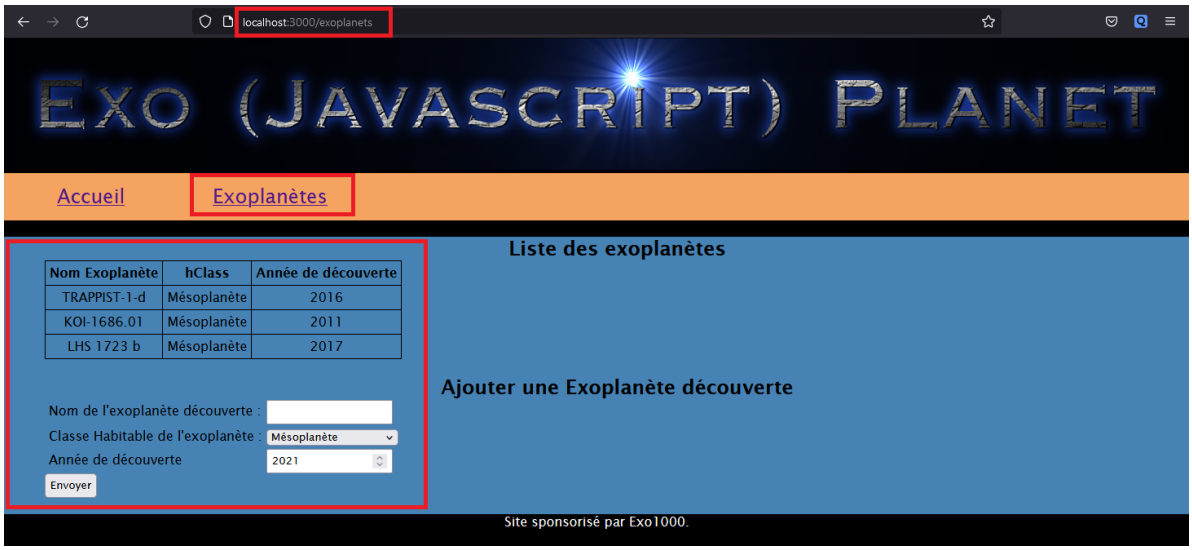
1. Il n'est pas nécessaire d'utiliser les volumes pour cet exercice.
2. Attention au format YAML qui nécessite une indentation homogène via des espaces (pas de tabulation !).
3. Il est utile de supprimer les conteneurs et images problématiques pour plus de lisibilité.

docker compose syllabusHTML

1. Vérifiez votre installation de docker-compose sur votre machine Debian
2. Créez une image Docker avec Apache et le site HTML (HTTP) via un docker-compose.yml
    a. Réutilisez ce que vous avez fait (Dockerfile) à la séance précédente
    b. Une solution de la séance précédente est disponible sur MooVin si vous n'avez pas votre solution
    c. Au niveau de la création des conteneurs, vous ferez un mapping de port (port forwarding) entre votre machine hôte (8091) et le conteneur (80). Ainsi le site HTML sera accessible depuis votre machine hôte sur le port 8091
3. Tester votre solution : lynx http://localhost:8091

Exercice 2 : Node.js & PgSQL

Déployer une application node.js basique utilisant une base de données PostgreSQL. L'application sera un site consacré au exoplanètes (voir ressources). Ce site affiche 3 exoplanètes et permet d'en ajouter.



Indications utiles

1. Vous avez dans les ressources un répertoire exoplanets_pgsql.zip. Décompressez ce zip sur votre VM Azure et commencez le travail directement dans ce répertoire.
2. Ce répertoire contient :
    a. Le code source du site (répertoire exoplanets)
    b. Un script de création de la db (répertoire initdb)
    c. Un fichier Dockerfile à remplir
    d. Un fichier docker-compose à remplir
3. Vous avez dans les ressources un fichier myappNodeJS.zip qui contient un exemple de déploiement d'une application Node.js sans base de données.
4. Vous avez dans les ressources un fichier sitePHPDockercompose.zip qui contient un exemple de déploiement d'un site PHP avec une base de données.
5. Quand on teste une image il est parfois nécessaire de tout nettoyer (conteneurs/images, volumes, réseau). La commande suivante fait cela très bien : docker-compose down –rmi all –volumes

Quand tout fonctionne, changez le nom du service « postgres_db » dans le fichier docker-compose.yml par le nom « data ». Pourquoi cela ne fonctionne plus ? Pouvez-vous corriger le souci en laissant « data » comme nom de service pour la db ?