**Tutoriel déploiement plateforme sur serveur**

La marche à suivre pour déployer la plateforme sur un serveur se déroule comme suit :

* Se munir des identifiants de connexion au serveur (adresse de l’hôte + mot de passe)
  + Exemple sur le serveur de développement
    - * Hôte : ns3011639.ip-149-202-69.eu
      * Nom d’utilisateur : aurelien
      * Mot de passe : aidee9oNgieth-ah
* Se munir des URLs de connexion au back et au front de la plateforme
  + Exemple sur le serveur de développement
    - * URL front : https://mapx-vigne-dev.servadmin.fr
      * URL back : <https://mxv-back-dev.servadmin.fr/>
* A l’aide des identifiants de connexion, se connecter à Putty et à Filezilla (dans les deux cas le port est le 22)
* Générer un build du front
  + Dans les fichiers **backend\_services.js** et **db\_interactions.js** (dossier « services ») s’assurer que les variables baseURL pointent vers l’adresse désirée selon qu’il s’agisse d’une utilisation en local ou d’un déploiement sur serveur
  + Dans un terminal, celui de Visual Studio Code par exemple, se placer dans le dossier principal du front (exemple «C:\Users\richaaur\Documents\Plateforme\MapX Vigne-20230830T134720Z-001\MapX Vigne> MapX Vigne ») et lancer la commande **npm run build**
  + Cette commande génère un dossier « dist » dans le dossier principal de front
* Import du front sur le serveur
  + Sur le serveur, créer un dossier dédié au frontend
  + Dans ce dossier, uploader le contenu du dossier « dist » généré à l’étape précédente
  + Le fichier index.html (présent dans le dossier « dist ») sert de point d’entrée à la plateforme via l’URL du front
* Import du back sur le serveur (3 conteneurs Docker)
  + Sur le serveur, créer un dossier dédié au backend
  + Dans ce dossier, uploader le contenu du dossier de backend. Ce dossier contient, entre autres, les scripts python assurant différents traitements sur les données, les fichiers d’initialisation de la base de données ainsi que divers fichiers de configuration docker, en particulier le fichier « docker-compose.yml » qui configure les différents conteneurs et permet de les lancer.
  + Note : il peut s’avérer nécessaire de modifier les ports définis dans le fichier « docker-compose.yml » si le lancement des conteneurs pose problème
* Lancement des conteneurs
  + Se placer dans le dossier du serveur dédié au backend (contenant le fichier « **docker-compose.yml** »)
  + Si le dossier « data » existe dans ce dossier, le supprimer via **putty** avec la commande **sudo rm -r data** pour que les containers soient entièrement reconstruits (il peut être nécessaire de stopper les containers en cours d’exécution au préalable pour cela)
  + Exécuter la commande **docker compose up -d**
  + Si tout se passe bien, les trois conteneurs sont lancés : scripts backend, base de données et phpMyAdmin
  + Note : la commande **docker ps -a** permet d’afficher les conteneurs actifs, **docker stop container\_id** permet de stopper un container, **docker rm container\_id** de supprimer un container
* Note : pour accéder à phpMyAdmin, se référer au fichier contenant les accès aux serveurs

Notes :

* Si d’autres conteneurs tournent déjà sur le serveur, il faut veiller à ce que les numéros de ports déclarés dans le fichier **docker-compose.yml** des nouveaux conteneurs soient différents de ceux des anciens. Ceci permet d’éviter que le lancement des nouveaux conteneurs écrase les anciens conteneurs. Pour visualiser la liste des conteneurs actifs, lancer la commande : **docker ps**
* Lorsque l’on ajoute un dossier depuis lequel on souhaite héberger le back ou le front d’une application, il est nécessaire de configurer/reconfigurer l’adressage permettant d’interroger les fichiers ou conteneurs se trouvant dans ce dossier. Pour cela, demander à l’administrateur du serveur de s’en charger en lui indiquant l’URL désirée et le dossier correspondant. Ici une URL fait le lien avec le dossier « front » et une autre avec le dossier « back »