

# Admin Infra : Labo 4

## 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 0 – Redimensionner la VM Azure

La VM Azure dispose de 0,5 GB de mémoire RAM. Ceci est insuffisant pour faire tourner Docker correctement. Nous allons donc redimensionner notre VM. Ceci est un avantage du Cloud, on parle **d'élasticité**. On peut facilement redimensionner notre VM c'est-à-dire ajouter de la mémoire, des disques, .... Evidemment cela n'est pas gratuit, le fournisseur Cloud (Azure ici) le fera contre paiement.

Nous allons donc redimensionner notre VM pour passer de 0,5 GB de mémoire à 1GB de mémoire. Pour ce faire :

1. Allez sur le site <https://portal.azure.com>
2. Sélectionnez votre VM
3. Assurez-vous que votre VM est éteinte
4. Cliquez sur **Taille**
5. Trier sur le coût par mois
6. Sélectionnez la formule B1s avec 1GB de RAM (8,20 €)
7. Cliquez sur le bouton **Redimensionner**

## Exercice 1 – Manipuler Docker

Nous continuons à construire l'infrastructure de notre société ITDEV. Celle-ci souhaite que ses développeurs utilisent des images docker pour leur développement afin de faciliter les déploiement, l'intégration continue et les tests.

**Référez-vous à la section 11.2.3 et 11.2.4 du syllabus. Lisez-bien les indications utiles ci-dessous avant de commencer l'exercice.**

### Indications utiles

1. Rappel des étapes pour cet exercice sont les suivantes : création du Dockerfile, création de l'image (docker build), création du conteneur (docker run).
2. Si vous avez des problèmes lors du « docker build », cela signifie que des erreurs sont présentes dans le Dockerfile. Vérifiez bien votre fichier (path, noms, ...)
3. Si vous avez des problèmes lors du « docker run » ou lors des tests, n'oubliez pas que vous pouvez vous connecter à l'intérieur du conteneur pour vérifier/tester différentes choses. Vous avez également une commande docker logs !
4. Il est utile de supprimer les conteneurs et images problématiques pour plus de lisibilité.

### Conteneurisation du site Apache syllabusHTML

1. Installer Docker sur votre machine Debian Azure
2. Créer un fichier Dockerfile
  - a. Celui sera basé sur l'image debian(latest)
  - b. Installez ensuite apache2
  - c. Vous pouvez copier un vhost déjà fait directement dans votre image de conteneur
    - i. Remplacer le contenu du fichier 000-default.conf présent dans /etc/apache2/sites-available par votre propre VirtualHost. Dans un conteneur nous ne déployons qu'un seul site donc autant que ce soit le site par défaut et de plus vous ne devrez pas l'activer (a2ensite).
  - d. Démarrer apache2
    - i. Attention par défaut un conteneur s'arrête lorsqu'il a effectué tous ses actions
    - ii. Regarder sur Internet pour que votre conteneur tourne en permanence. Ainsi Apache pourra répondre aux requêtes HTTP. Rechercher «How to start apache2 automatically in docker »
3. Créer une image Docker sur base de votre Dockerfile

4. Lancer un conteneur à partir de votre image Docker
  - a. Vous ferez un mapping de port (port forwarding) entre votre machine hôte(8090) et le conteneur (80).
  - b. Ainsi le site HTML sera accessible depuis votre machine hôte sur le port 8090
5. Tester le résultat sur votre machine Debian Azure : « lynx http://localhost:8090 »
6. Tester le résultat sur votre machine physique depuis un navigateur.
  - a. Cela ne marche pas ?
    - i. Qu'est-ce qui pourrait bloquer ?
    - ii. Quel est le chemin réseau pour accéder à mon conteneur Docker se trouvant dans ma VM Azure depuis ma machine physique ?

## Exercice 2 – Conteneuriser une simple application ASP.NET Core

Pour cet exercice, on vous demande de conteneuriser une simple application ASP.NET Core. Vous trouverez cette application dans les ressources sur MooVin sous le nom WebApplication2. Cette application permet d'afficher une liste d'items (vide au début) et d'ajouter des items à cette liste.

Pour cet exercice : à vous de faire vos preuves en étant moins guidé !

Pour réaliser la conteneurisation :

1. Essayez tout d'abord l'application avec Visual Studio et remplissez le fichier Dockerfile
2. N'hésitez pas à jeter un œil sur Internet
  - a. <https://learn.microsoft.com/en-us/dotnet/core/docker/build-container?tabs=linux>
3. Comprenez ce que vous faites/expérimentez et n'hésitez pas à poser des questions
4. La version de .NET utilisée est 3.1 (LTS)
5. docker logs peut vous aider !