	BTS SIO	
	Services Informatiques aux Organisations	
	Option	SISR
	Session	2021/22

Nom et prénom : Fabien CHEVALIER	Activité professionnelle N°	10
----------------------------------	------------------------------------	----

NATURE DE L'ACTIVITE	Mise en place d'un serveur LAMP conteneurisé avec Docker
Contexte	Atelier en classe
Objectifs	Déployer simplement et automatiquement un serveur web linux
Lieu de réalisation	H3-Campus

SOLUTIONS ENVISAGEABLES	
-	Utiliser docker et docker-compose pour déployer plusieurs conteneurs nécessaires à la mise en place d'un environnement de développement web classique

DESCRIPTION DE LA SOLUTION RETENUE	
Conditions initiales	Machine virtuelle Debian
Conditions finales	Environnement de développement web fonctionnel
Outils utilisés	Debian, Docker

CONDITIONS DE REALISATION	
Matériels	Macbook Pro 13'
Logiciels	VmWare Fusion
Durée	2h
Contraintes	

COMPETENCES MISES EN OEUVRE POUR CETTE ACTIVITE PROFESSIONNELLE	
Code	Intitulé
Gérer le patrimoine informatique	▸ Vérifier les conditions de la continuité d'un service informatique
Répondre aux incidents et aux demandes d'assistance et d'évolution	▸ Traiter des demandes concernant les services réseau et système, applicatifs
Mettre à disposition des utilisateurs un service informatique	▸ Réaliser les tests d'intégration et d'acceptation d'un service ▸ Déployer un service

DEROULEMENT DE L'ACTIVITE
I/ Présentation Docker permet de conteneuriser des applications, leur permettant par la suite d'être facilement déployable. Docker-compose permet d'automatiser via l'édition d'un petit script le lancement de plusieurs conteneurs à la fois et d'ainsi avoir à disposition son infrastructure fonctionnant sur n'importe quel système Linux.
II/ Installation de docker sous Debian L'installation de docker se déroule comme ceci :

```
Debian 11.x 64 bits
21 janv. 20:06
fabien@debian: ~
fabien@debian:~$ su -
Mot de passe :
root@debian:~# curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
root@debian:~# echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
root@debian:~# apt-get update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Réception de :4 https://download.docker.com/linux/debian bullseye InRelease [43,3 kB]
Réception de :5 https://download.docker.com/linux/debian bullseye/stable amd64 Packages [7 436 B]
50,8 ko réceptionnés en 0s (111 ko/s)
Lecture des listes de paquets... Fait
root@debian:~# apt-get install docker-ce docker-ce-cli containerd.io
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl
  libslirp0 patch pigz slirp4netns
Paquets suggérés :
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
  ed diffutils-doc
Les NOUVEAUX paquets suivants seront installés :
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin git git-man liberror-perl libslirp0 patch pigz
  slirp4netns
0 mis à jour, 12 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 105 Mo dans les archives.
Après cette opération, 447 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 pigz amd64 2.6-1 [64,0 kB]
Réception de :2 http://deb.debian.org/debian bullseye/main amd64 liberror-perl all 0.17029-1 [31,0 kB]
Réception de :3 http://deb.debian.org/debian bullseye/main amd64 git-man all 1:2.30.2-1 [1 827 kB]
Réception de :4 https://download.docker.com/linux/debian bullseye/stable amd64 containerd.io amd64 1.4.12-1 [23,7 MB]
Réception de :5 http://deb.debian.org/debian bullseye/main amd64 git amd64 1:2.30.2-1 [5 527 kB]
Réception de :6 http://deb.debian.org/debian bullseye/main amd64 libslirp0 amd64 4.4.0-1+deb11u2 [57,9 kB]
Réception de :7 http://deb.debian.org/debian bullseye/main amd64 patch amd64 2.7.6-7 [128 kB]
Réception de :8 http://deb.debian.org/debian bullseye/main amd64 slirp4netns amd64 1.0.1-2 [33,4 kB]
Réception de :9 https://download.docker.com/linux/debian bullseye/stable amd64 docker-ce-cli amd64 5:20.10.12-3-0-debian-bullseye [40,7 MB]
Réception de :10 https://download.docker.com/linux/debian bullseye/stable amd64 docker-ce amd64 5:20.10.12-3-0-debian-bullseye [21,2 MB]
Réception de :11 https://download.docker.com/linux/debian bullseye/stable amd64 docker-ce-rootless-extras amd64 5:20.10.12-3-0-debian-bullseye [7 91
9 kB]
Réception de :12 https://download.docker.com/linux/debian bullseye/stable amd64 docker-scan-plugin amd64 0.12.0-debian-bullseye [3 518 kB]
105 Mo réceptionnés en 3s (40,6 Mo/s)
Sélection du paquet pigz précédemment désélectionné.
(Lecture de la base de données... 143317 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de :/00-pigz 2.6-1 amd64 deb
```

Ensuite, on installe docker-compose :

```
root@debian:~# sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/do
cker-compose
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 664    100 664    0     0  9485      0 --:--:-- --:--:-- --:--:--  9485
100 12.1M 100 12.1M    0     0 16.0M      0 --:--:-- --:--:-- --:--:-- 31.1M
root@debian:~# chmod +x /usr/local/bin/docker-compose
root@debian:~# exit
déconnexion
fabien@debian:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
fabien@debian:~$
```

III/ Préparation du fichier docker-compose.yml

Docker-compose se base sur un fichier appelé docker-compose.yml pour configurer les conteneurs et leurs interactions. Voici le fichier utilisé (je l'ai commenté pour expliquer le fonctionnement) :

```
Ouvrir  *docker-compose.yml  Enregistrer  x
~/Documents/docker

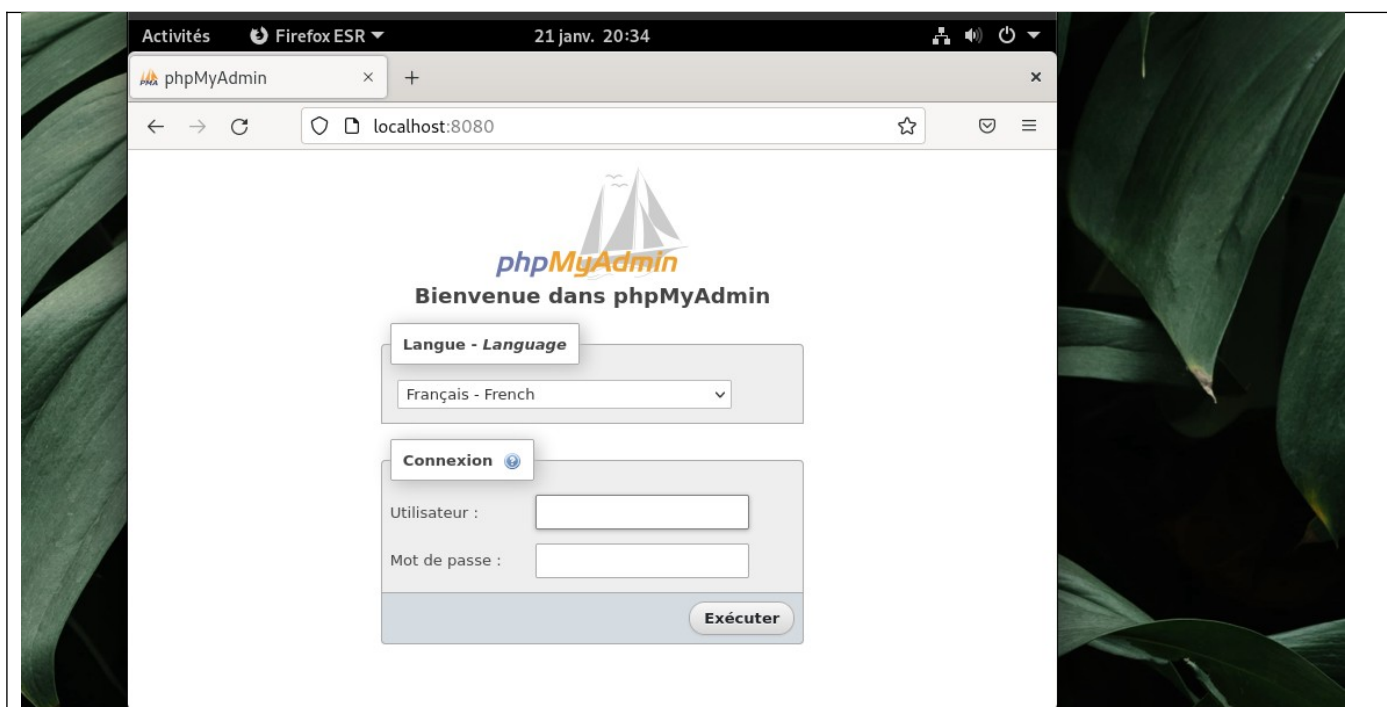
1 version: '2'
2
3 services:
4   web: #conteneur apache/php
5     image: lavoweb/php-5.6
6     ports:
7       - "80:80"
8     volumes:
9       - ~/.docker/www:/var/www/html
10    links:
11      - db:db
12   db: #conteneur MySQL
13     image: mysql:5.5
14     volumes:
15       - ~/.docker/mysql:/var/lib/mysql
16     ports:
17       - "3306:3306"
18     environment:
19       - MYSQL_ROOT_PASSWORD=root
20   myadmin: #conteneur phpmyadmin
21     image: phpmyadmin/phpmyadmin
22     ports:
23       - "8080:80"
24     links:
25       - db:db

YAML  Largeur des tabulations : 8  Lig 20, Col 35  INS
```

IV/ Lancement de notre serveur web lamp

```
[2100] Failed to execute script docker-compose
fabien@debian:~/Documents/docker$ su -
Mot de passe :
root@debian:~# ls
root@debian:~# cd /home/fabien
root@debian:/home/fabien# ls
Bureau Documents Modèles Public Vidéos
core Images Musique Téléchargements
root@debian:/home/fabien# cd Documents/
root@debian:/home/fabien/Documents# ls
docker
root@debian:/home/fabien/Documents# cd docker/
root@debian:/home/fabien/Documents/docker# sl
-bash: sl : commande introuvable
root@debian:/home/fabien/Documents/docker# ls
docker-compose.yml
root@debian:/home/fabien/Documents/docker# docker-compose up
Creating network "docker_default" with the default driver
Pulling db (mysql:5.5)...
5.5: Pulling from library/mysql
743f2d6c1f65: Pull complete
3f0c413ee255: Pull complete
aef1ef8f1aac: Pull complete
f9ee573e34cb: Pull complete
3f237e01f153: Pull complete
03dale065b16: Pull complete
04087a801070: Pull complete
7efd5395ab31: Pull complete
1b5cc03aac8: Pull complete
2b7adaec9998: Pull complete
385b8f96a9ba: Pull complete
Digest: sha256:12da85ab88aedfdf39455872fb044f607c32fdc233cd59f1d26769fbf439b045
Status: Downloaded newer image for mysql:5.5
Pulling web (lavoweb/php-5.6)...
[2100] Failed to execute script docker-compose
```

Il suffit de taper la commande `docker-compose up` depuis le dossier où se trouve le fichier `docker-compose.yml`. On peut tester le bon fonctionnement de notre serveur en se rendant à l'adresse `localhost :8080` :



C'est aussi simple que cela.

CONCLUSION

Docker couplé à docker-compose permet de configurer de façon extrêmement simple des environnements de production automatisés.

EVOLUTION POSSIBLE

Il est possible de configurer plus précisément le déploiement, pour permettre par exemple d'installer un logiciel comme GLPI très rapidement. Pour cela, il existe des paramètres à ajouter dans la configuration de MySQL ou Apache afin de rendre possible une telle chose.