

Petite récapitulation pour la partie Docker  
Epsi Paris  
Décembre 2019  
Y. STROPPA

Pour ceux qui dormaient pendant la présentation, je vous rappelle que Docker s'impose dans le monde Devops de plus en plus et permet de rendre des services difficilement remplaçables par d'autres solutions. De plus les systèmes d'orchestration sont omniprésents dans les architectures en mode production ... D'où le réveil de chacun face à cette technologie ...

Dans nos affaires, il suffit pour l'instant de monter trois conteneurs assez simple à faire ...

Rappel les images Docker sont des File System que l'on peut se procurer à partir du site de Docker Hub .... sur vos machines vous pouvez configurer une VM avec un Ubuntu 18,04 ou un centos 7. Par file System on entend un système de fichiers, une arborescence de fichiers qui a la particularité d'être constituée à partir de l'UFS (Union File System). Ce qui veut dire qu'une image est un ensemble de couches superposées et qui constituent votre image.

Quelques commandes usuelles pour commencer :

Pour vérifier les images disponibles sur le hub il suffit de faire une recherche à l'aide de la commande suivante :

**#docker search ubuntu**  
**#docker search nodejs**

ou autre recherche en fonction de vos besoins

Le résultat de cette commande est une liste d'image certifiées par Docker que vous pouvez utiliser....

Pour que docker fonctionne sur vos machines, il vous faut au préalable effectuer l'installation de Docker et surtout démarrez le service Docker engine ..... qui va permettre de gérer les conteneurs sur votre machine.

Sur Windows : on pourra installer en local Kitematic qui regroupe toutes les fonctionnalités associées à Docker .... Petit rappel quand même n'oubliez pas que les images sont OS dépendantes ... ce n'est pas une VM que l'on installe mais un File System sur un Hôte dans lequel on exécute un service ...

Sur Linux : `systemctl status docker`

- `docker.service` - Docker Application Container Engine  
Loaded: loaded (/lib/systemd/system/docker.service; disabled; vendor preset: enabled)  
Active: active (running) since Mon 2019-12-09 08:45:49 CET; 2 days ago  
Docs: <https://docs.docker.com>  
Main PID: 2977 (dockerd)  
Tasks: 63

Ensuite il faut que votre compte soit autorisé à utiliser docker ... pour ce faire veuillez vérifier si le compte y accède à l'aide de la commande

```
#docker ps
```

Si votre système affiche ce contenu c'est que tout est OK.

#docker ps				
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			

C'est que vous avez le droit d'utiliser docker, dans le cas contraire il vous informe que vous essayez d'accéder à /var/run/docker.sock et que vous n'êtes pas autorisé ... dans ce cas suivre les procédures indiquées sur le Web pour autoriser votre compte ... généralement il suffit d'affecter le compte au groupe Docker et de rebooter votre Vm ...

Une fois les problèmes de droit réglés ... vous pouvez commencer à utiliser en testant votre système par l'utilisation d'une image de démonstration qui est busybox et la commande suivante :

```
#docker run busybox
```

Ce que fait Docker, c'est dans un premier temps vérifier que l'image n'est pas dans votre dépôt local et la télécharger. Pour vérifier sur votre image a bien été téléchargée ... exécutez la commande #docker images

Vous devez visualiser dans la liste des images de votre dépôt local l'image en question ...

Nos installations de conteneur :

Dans notre projet on se propose d'installer 3 conteneurs :

- Mongo
- Mysql
- Sqlserver

Pour ce faire, il suffit de faire un pull de ces trois images et ensuite on va configurer leur exécution ...

Le pull va uniquement redescendre l'image dans votre dépôt local :

```
docker pull mongo
docker pull mysql
docker pull exoplatform/sqlserver
```

On vérifiera dans notre dépôt de la présence de ces trois images ...

Ensuite pour exécuter les conteneurs il faut s'occuper de deux choses importantes c'est de savoir quel est le volume de l'image afin de pouvoir préserver les données que l'on va stocker dans ces images ...

Pour mongodb il suffit de regarder et de faire un inspect sur l'image

#### **#docker inspect mongo**

```
"Volumes": {  
  "/data/configdb": {},  
  "/data/db": {}  
},  
"WorkingDir": "",
```

```
"ExposedPorts": {  
  "27017/tcp": {}  
},
```

#### **#docker inspect mysql**

```
"Volumes": {  
  "/var/lib/mysql": {}  
},  
"WorkingDir": "",
```

```
"ExposedPorts": {  
  "3306/tcp": {},  
  "33060/tcp": {}  
},
```

#### **#docker inspect exoplatfrom/sqlserver**

```
"Volumes": null,  
"WorkingDir": "/sqlserver",
```

```
"ExposedPorts": {  
  "1433/tcp": {}  
},
```

Donc maintenant on a les volumes utilisés et le ports d'exposition de ces services ...

Pour démarrer ces services on va exécuter les commandes suivantes dans votre environnement :

Pour mongo :

```
#docker run -p 27017:27017 -v /home/ystroppa/epsi/DOTNET/mongo:/db/data -d mongo
```

Pour mysql

```
#docker run -e MYSQL_ROOT_PASSWORD=rmc2012 -p 3306:3306 -v /home/ystroppa/epsi/DOTNET/mysql:/var/lib/mysql -d mysql
```

Pour sqlserver

```
#docker run -d -v /home/ystroppa/epsi/DOTNET/sqlserver:/sqlserver -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=RjBnc450" -e SQLSERVER_USER="yvan" -e SQLSERVER_PASSWORD="yvan" -e SQLSERVER_DATABASE=epsi -p 1433:1433 --name sql3 exoplatfrom/sqlserver'
```

Pour vérifier que ces services sont bien démarrés : il suffit de taper la commande suivante :

**#docker ps**

Vous devez visualiser les services docker

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
8028519546dc	exoplatform/sqlserver	"/entrypoint.sh"	45 hours ago	Up 45 hours	0.0.0.0:1433->1433/tcp
f2647a2aafcc	mongo	"docker-entrypoint.s..."	4 weeks ago	Up 2 days	0.0.0.0:27018->27017/tcp
6409235c3114	mysql:5.7	"docker-entrypoint.s..."	4 weeks ago	Up 2 days	0.0.0.0:3306->3306/tcp, 33060/tcp

L'important c'est que les services sont bien actifs ... et que les ports sont accessibles

0.0.0.0:3306 → 3306

0.0.0.0:27017 → 27017

0.0.0.0:1433 → 1433

Sur un système linux ou Mac os on pourra à l'aide de la commande netstat -a vérifiez que ces services sont bien en mode Listen ...

Si vous voulez tester l'accessibilité de ces services, vous pouvez installer un client mysql, mongo ou sqlserver et tester la connectivité ...

Avec un client mongo

**#mongo -h 127.0.0.1**

Avec un client mysql

**#mysql -h 127.0.0.1 -u root -p**

Avec un client sqlserver

**#sqlcmd -S 127.0.0.1 -U SA -P RjBnc450**

Une fois que tout cela fonctionne, vous êtes prêts à développer vos différentes parties en C# et vérifiez les interactions entre ces différentes parties applicatif et base de données.

Pour les parties Mysql et sqlserver on reste dans le domaine de la base de données Relationnelle .. avec des commandes SQL pour le LDD et le LMD ....pour la partie Mongo on est dans du NOSQL en mode documents.

Ici on manipule dans des databases, des collections de documents sous formats JSON ....

On pourra utiliser des outils graphiques de type robo3T qui permettent d'avoir un visualiser plus facile pour l'interprétation du contenu ...

Bon courage

