

**Module: Conteneurisation Année Universitaire: 2022/2023**

*Responsable du module : Missaoui Abdelbaki*

**Prérequis :**

- Installer VirtualBox sur la machine hôte

**Installation de Docker-machine :**

Il faut avoir installé docker au préalable puis ensuite passer les commandes suivantes :

```
base=https://github.com/docker/machine/releases/download/v0.14.0 &&  
curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-machine &&  
sudo install /tmp/docker-machine /usr/local/bin/docker-machine
```

**Création d'un manager Swarm :**

```
docker-machine create --virtualbox-hostonly-cidr 192.168.61.1/24 --driver virtualbox  
swarm-manager
```

**Création de 2 Workers Swarm :**

```
docker-machine create --virtualbox-hostonly-cidr 192.168.61.1/24 --driver virtualbox  
swarm-worker1  
docker-machine create --virtualbox-hostonly-cidr 192.168.61.1/24 --driver virtualbox  
swarm-worker2
```

**Vérification de l'existence des Swarm :**

```
docker-machine ls
```

**Accès sur le manager Swarm :**

```
docker-machine ssh swarm-manager
```

**Le déclarer manager officiellement :**

```
docker-machine ssh swarm-manager "docker swarm init --advertise-addr 192.168.61.100"  
exit
```

Une fois le manager déclaré il faut enregistrer le token qui nous ai donné après avoir effectué la commande dans un fichier .txt :

```
sam@sam:~$ docker-machine ssh swarm-manager "docker swarm init --advertise-addr  
192.168.61.100"
```

Swarm initialized: current node (unekzgdabprv9l7qtw6nnh1n4) is now a manager.

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-427lnhaopn72xqpluwjn1yphtxj6facak08zzh67mhad215xsj-2hr28c8ga5hplvigh2bd5arpy192.168.61.100:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

#### Accès sur Worker1 et Worker2 :

```
docker-machine ssh swarm-worker1  
exit
```

#### Accès sur Worker2 :

```
docker-machine ssh swarm-worker2  
exit
```

#### Ajouter un travailleur Worker1 à un swarm-manager :

```
docker-machine ssh swarm-worker1 "docker swarm join --token SWMTKN-1-427lnhaopn72xqpluwjn1yphtxj6facak08zzh67mhad215xsj-2hr28c8ga5hplvigh2bd5arpy192.168.61.100:2377"
```

#### Ajouter un travailleur Worker2 à un swarm-manager :

```
docker-machine ssh swarm-worker2 "docker swarm join --token SWMTKN-1-427lnhaopn72xqpluwjn1yphtxj6facak08zzh67mhad215xsj-2hr28c8ga5hplvigh2bd5arpy192.168.61.100:2377"
```

#### Voyons les nœuds qui tournent :

```
docker-machine ssh swarm-manager "docker node ls"
```

#### Infos sur les Conteneurs en dehors du Manager :

```
docker ps -a
```

#### Infos sur les Conteneurs dans le Manager :

```
docker-machine ssh swarm-manager "docker ps -a"
```

#### Configuration de variables d'environnements afin de pouvoir créer des conteneurs dans le Swarm mais depuis l'extérieur (sans avoir besoin de faire un ssh) :

```
docker-machine env swarm-manager  
eval $(docker-machine env swarm-manager)  
docker-machine active
```

**Téléchargement d'une image depuis Docker Hub :**

```
docker pull alpine
```

**Création de 5 conteneurs Docker de cette image dont le nom est shark-app et les ports (9000:9091) :**

```
docker service create --name shark-app --replicas 5 --entrypoint "tail -f /dev/null" --publish published=9000,target=9091 --restart-condition=on-failure --limit-memory 100M alpine
```

**Création de 2 conteneurs Docker de nginx (dont le nom est shark-nginx) et les ports (9001:9091) :**

```
docker service create --name shark-nginx --replicas 2 --publish published=9001,target=9091 --restart-condition=on-failure --limit-memory 100M nginx
```

**Vérification de l'existence des Swarm :**

```
docker service ls  
exit
```

**Accès via un navigateur (ex:firefox) ou via curl sur shark-nginx :****Scale shark-nginx en 3 replicas :**

```
docker service scale shark-nginx=3
```

**Vérification d'existence des Swarm :**

```
docker service ps shark-nginx
```

**Monter un Volume :**

```
docker service create --replicas 1 -p 8092:80 --mount type=bind,source=/website,target=/usr/share/nginx/html --name volumeweb nginx
```

**Autres commandes utiles :****1- Mise à jour d'un service :**

```
docker service update --image nomservice:version nomconteneur  
(ex nginx:1.4 shark-nginx)
```

**2- Arrêter le noeud worker1 :**

```
docker node update --availability drain worker1
```

**3- Quitter et sortir proprement :**

```
docker swarm-manager leave  
docker-machine stop swarm-manager
```

```
docker-machine rm swarm-manager
```