

Docker Swarm

Exercice 1 : Exercice avec un visualiseur et les Labels

Un label permet d'associer une étiquette (un tag) à un node ou un service. Ceci permet par exemple de déployer des services sur un ou des node(s) ayant cette étiquette.

Un visualiseur web basique :

```
docker service create --name=viz --publish=8080:8080 \
  --constraint=node.role==manager \
  --mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
  dockersamples/visualizer
```

Consulter l'url : http://ip_master:8080

Créer un service de nginx nommé web en associant le port 80 du hôte avec le port 80 du conteneur.

Consulter l'url : http://ip_master:8080

Activer deux instances du service web.

Consulter l'url : http://ip_master:8080

Tester le failover en arrêtant la machine worker2.

Consulter l'url : http://ip_master:8080

Démarrer la machine worker2.

Consulter l'url : http://ip_master:8080

Activer une seule instance de web.

Consulter l'url : http://ip_master:8080

Supprimer le service web.

Manipuler un label :

Mettre à jour le node worker2 en lui associant un label intitulé "mars".

```
docker node update --label-add nodelabel=mars worker2
```

Inspecter les informations du node worker2.

```
docker inspect worker2
```

Démarrer un service de nginx nommé web avec la contrainte d'être sur un node ayant le tag "mars".

```
docker service create --name web \  
    --constraint node.labels.nodelabel==mars \  
    nginx
```

Vérifier que le service fonctionne sur un node taggé "mars".

```
docker service ps web
```

Consulter l'url : http://ip_master:8080

Activer trois instances du service web.

Vérifier où fonctionne ces services.

Consulter l'url : http://ip_master:8080

Supprimer le service web.

Supprimer le service du visualiseur.

Supprimer le label "mars" du node worker2.

Exercice 2 : Exercice utilisant des fichiers yaml et des stacks, Update d'une stack

Un fichier yaml pour un visualiseur web :

Soit la commande suivante :

```
docker service create --name=viz -p 8080:8080  
--constraint=node.role==manager \  
-v /var/run/docker.sock=/var/run/docker.sock \  
dockersamples/visualizer
```

Créer un fichier yaml viz.yml pour réaliser l'équivalent de la commande ci-dessus.
Ce fichier sera lancé en tant que stack.

Exécuter une stack nommée viz en utilisant le fichier viz.yml.

Exécuter les commandes suivantes :

```
docker service ls  
docker service ps viz_viz  
docker stack ls  
docker stack ps viz  
docker stack services viz
```

Consulter l'url : http://ip_master:8080

Création d'une stack web :

Créer un fichier yaml web.yml qui sera lancé en tant que stack. Deux services seront lancés :

- un service de nginx nommé web1, le port 8081 du hôte est mappé sur le port 80 du service.
- un service de httpd nommé web2, le port 8082 du hôte est mappé sur le port 80 du service.

Exécuter une stack nommée web en utilisant le fichier web.yml.

Vérifier en listant les stacks.

Consulter l'url : http://ip_master:8080

L'update d'une stack :

Soit le fichier web2.yml suivant :

```
version: '3.1'

services:
  web1:
    image: nginx
    ports:
      - 8081:80
    deploy:
      replicas: 3
  web2:
    image: httpd
    ports:
      - 8082:80
    deploy:
      replicas: 2
```

Créer ce fichier web2.yml, et mettre à jour la stack web en utilisant ce fichier.

Vérifier avec les commandes suivantes :

```
docker stack ls
docker service ls
```

Consulter l'url : http://ip_master:8080

Ainsi le mot clé “deploy” sert aussi bien au déploiement initial d'une stack qu'à sa mise à jour.

Supprimer les stacks.

Exercice 3 : Exercice utilisant les utilitaires web visualiser et portainer

Créer le fichier monitoring.yml suivant :

```
version: '3'

services:

  portainer:
    image: portainer/portainer
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    ports:
      - 9000:9000
    deploy:
      placement:
        constraints: [node.role == manager]

  viz:
    image: dockersamples/visualizer
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    deploy:
      placement:
        constraints: [node.role == manager]
    ports:
      - 9090:8080
```

Exécuter une stack nommé monitoring en utilisant le fichier monitoring.yml.

Vérifier avec la commande suivante :

```
docker stack services monitoring
```

Consulter l'url de visualizer : http://ip_master:9090

Consulter l'url de portainer : http://ip_master:9000

De la page web :

- créer le compte en saisissant un mot de passe (admin1234)
- puis cliquer sur le premier “carré” : LOCAL
- visiter les menus disponibles.

- créer une stack nommée web avec :

```
version: '3.1'

services:
  web1:
    image: nginx
    ports:
      - 8081:80
    deploy:
      replicas: 3
  web2:
    image: httpd
    ports:
      - 8082:80
    deploy:
      replicas: 2
```

- consulter les menus pour “explorer” et tester la stack web.

Lorsque vous aurez terminé, supprimer les stacks.

Correction - Docker Swarm

Exercice 1 : Exercice avec un visualiseur et les Labels

Un label permet d'associer une étiquette (un tag) à un node ou un service. Ceci permet par exemple de déployer des services sur un ou des node(s) ayant cette étiquette.

Un visualiseur web basique :

```
docker service create --name=viz --publish=8080:8080 \  
  --constraint=node.role==manager \  
  --mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \  
  dockersamples/visualizer
```

Consulter l'url : http://ip_master:8080

Créer un service de nginx nommé web en associant le port 80 du hôte avec le port 80 du conteneur.

```
docker service create --name=web --publish=80:80 nginx
```

Consulter l'url : http://ip_master:8080

Activer deux instances du service web.

```
docker service scale web=2
```

Consulter l'url : http://ip_master:8080

Tester le failover en arrêtant la machine worker2.

```
docker-machine stop worker2           pour le failover
```

Consulter l'url : http://ip_master:8080

Démarrer la machine worker2.

```
docker-machine start worker2
```

Consulter l'url : http://ip_master:8080

Activer une seule instance de web.

```
docker service scale web=1
```

Consulter l'url : http://ip_master:8080

Supprimer le service web.

```
docker service rm web
```

Manipuler un label :

Mettre à jour le node worker2 en lui associant un label intitulé "mars".

```
docker node update --label-add nodelabel=mars worker2
```

Inspecter les informations du node worker2.

```
docker inspect worker2
```

Démarrer un service de nginx nommé web avec la contrainte d'être sur un node ayant le tag "mars".

```
docker service create --name web \  
    --constraint node.labels.nodelabel==mars \  
    nginx
```

Vérifier que le service fonctionne sur un node taggé "mars".

```
docker service ps web
```

Consulter l'url : http://ip_master:8080

Activer trois instances du service web.

```
docker service scale=3 web
```

Vérifier où fonctionne ces services.

```
docker service ps web
```

Consulter l'url : http://ip_master:8080

Supprimer le service web.

```
docker service rm web
```

Supprimer le service du visualiseur.

```
docker service rm viz
```

Supprimer le label "mars" du node worker2.

Exercice 2 : Exercice utilisant des fichiers yaml et des stacks, Update d'une stack

Un fichier yaml pour un visualiseur web :

Soit la commande suivante :

```
docker service create --name=viz -p 8080:8080  
--constraint=node.role==manager \  
-v /var/run/docker.sock=/var/run/docker.sock \  
dockersamples/visualizer
```

Créer un fichier yaml viz.yml pour réaliser l'équivalent de la commande ci-dessus.
Ce fichier sera lancé en tant que stack.

Le fichier viz.yml :

```
version: "3.1"  
  
services:  
  viz:  
    image: dockersamples/visualizer  
    ports:  
      - 8080:8080  
    volumes:  
      - /var/run/docker.sock:/var/run/docker.sock  
    deploy:  
      placement:  
        constraints: [node.role == manager]
```

Exécuter une stack nommée viz en utilisant le fichier viz.yml.

```
docker stack deploy -c viz.yml viz
```

Exécuter les commandes suivantes :

```
docker service ls  
docker service ps viz_viz  
docker stack ls  
docker stack ps viz  
docker stack services viz
```

Consulter l'url : http://ip_master:8080

Création d'une stack web :

Créer un fichier yaml web.yml qui sera lancé en tant que stack. Deux services seront lancés :

- un service de nginx nommé web1, le port 8081 du hôte est mappé sur le port 80 du service.
- un service de httpd nommé web2, le port 8082 du hôte est mappé sur le port 80 du service.

Le fichier web.yml

```
version: '3.1'

services:
  web1:
    image: nginx
    ports:
      - 8081:80
  web2:
    image: httpd
    ports:
      - 8082:80
```

Exécuter une stack nommée web en utilisant le fichier web.yml.

```
docker stack deploy -c web.yml web
```

Vérifier en listant les stacks.

```
docker stack ls
```

Consulter l'url : http://ip_master:8080

L'update d'une stack :

Soit le fichier web2.yml suivant :

```
version: '3.1'

services:
  web1:
    image: nginx
    ports:
      - 8081:80
    deploy:
      replicas: 3
  web2:
    image: httpd
    ports:
      - 8082:80
    deploy:
      replicas: 2
```

Créer ce fichier web2.yml, et mettre à jour la stack web en utilisant ce fichier.

```
docker stack deploy -c web2.yml web
```

Vérifier avec les commandes suivantes :

```
docker stack ls  
docker service ls
```

Consulter l'url : http://ip_master:8080

Ainsi le mot clé “deploy” sert aussi bien au déploiement initial d'une stack qu'à sa mise à jour.

Supprimer les stacks.

```
docker stack rm web viz
```

Exercice 3 : Exercice utilisant les utilitaires web visualiser et portainer

Créer le fichier monitoring.yml suivant :

```
version: '3'

services:

  portainer:
    image: portainer/portainer
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    ports:
      - 9000:9000
    deploy:
      placement:
        constraints: [node.role == manager]

  viz:
    image: dockersamples/visualizer
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    deploy:
      placement:
        constraints: [node.role == manager]
    ports:
      - 9090:8080
```

Exécuter une stack nommé monitoring en utilisant le fichier monitoring.yml.

```
docker stack deploy -c monitoring.yml monitoring
```

Vérifier avec la commande suivante :

```
docker stack services monitoring
```

Consulter l'url de visualizer : http://ip_master:9090

Consulter l'url de portainer : http://ip_master:9000

De la page web :

- créer le compte en saisissant un mot de passe (admin1234)
- puis cliquer sur le premier “carré” : LOCAL
- visiter les menus disponibles.
- créer une stack nommée web avec :

```
version: '3.1'

services:
```

```
web1:
  image: nginx
  ports:
    - 8081:80
  deploy:
    replicas: 3
web2:
  image: httpd
  ports:
    - 8082:80
  deploy:
    replicas: 2
```

- consulter les menus pour “explorer” et tester la stack web.

Lorsque vous aurez terminé, supprimer les stacks :

```
docker stack rm web monitoring
```