TP réseau

Gestion des réseaux

Création de réseaux

La création d'un réseau se fait via la commande suivante :

```
docker network create [options] network-name
```

Création du réseau iut-1

Il est possible de configurer le réseau, notamment le driver à utiliser, la plage d'IP à utiliser, le sous-réseau depuis lequel seront sélectionnées les IPs ou encore si le réseau doit être interne.

- En vous basant sur l'aide de la commande (`docker network --help`), créer un réseau :
    - nommé **iut-internal ;**
    - **interne** (pas d'accès vers l'extérieur) ;
    - utilisant le **sous-réseau 10.0.1.0/28 ;**
    - utilisant le driver **bridge**.

Création du réseau iut-internal-2

- Créer le réseau :
    - nommé **iut-internal-2** ;
    - ayant un accès à l'extérieur (Internet) ;
    - utilisant le sous-réseau **10.0.1.0/29** ;
    - utilisant le driver **bridge**.

- Est-ce que cette création est possible ? Si non, pourquoi ?

Liste des réseaux de la machine

Afficher la liste des réseaux de la machine hôte :

```
docker network list
```

Inspection d'un réseau

```
docker network inspect
```

L’inspection d’un réseau permet de récupérer certaines informations importantes.

- citer quelques informations pertinentes ;

Network Drivers

- Via la commande `docker info`, lister les drivers réseaux disponibles.

Communication entre conteneurs (ping)

Attacher un réseau à un conteneur permet à ce conteneur de pouvoir communiquer avec tous les conteneurs y étant reliés.
Les conteneurs peuvent se joindre entre eux via leurs noms. La résolution DNS est effectuée par Docker.

```
docker network connect iut-internal-2 container-name
```

1. Créer deux conteneurs (dans deux terminaux différents) :

```
docker run -it --name ping-dst willfarrell/ping
docker run -it --name ping-src -e HOSTNAME=ping-dst -e TIMEOUT=1
willfarrell/ping
```

- Quel est le rôle de l’image utilisée ?
- Qu’observe-t-on dans le conteneur « ping-src » ?

2. Tout en laissant les deux conteneurs créé fonctionner, créer un réseau et y ajouter les conteneurs nouvellement créés :

```
docker network create ping-net
docker network connect ping-net ping-dst
docker network connect ping-net ping-src
```

- Qu’observe-t-on dans le conteneurs « ping-src » ?

3. Stopper, supprimer et recréer le second conteneur avec l’argument supplémentaire `--link` :

```
docker run -it --name ping-src --link ping-dst -e HOSTNAME=ping-dst -e
TIMEOUT=1 willfarrell/ping
```

- Cette fois, que peut-on observer ?
- D'après la documentation docker, à quoi sert ce paramètre **--link** ?
- D'après la documentation docker, doit-on utiliser ce paramètre ?

Étude des interfaces réseaux du conteneur et du système hôte

1. Inspecter le conteneur **ping-dst** (`docker inspect`). On remarque que deux réseaux sont liés au conteneur. L'un est celui que nous venons d'ajouter ; À votre avis, quel est le second et quel est son rôle ?

2. Dans le conteneur, afficher les interfaces (via un nouveau terminal) :

```
docker exec ping-src ip link
```

- Combien d'interfaces sont présentes ?
- Comparer les adresses de ces interfaces (`docker exec ping-src ip address`) avec les adresses des interfaces de la machine hôte (`ip address`). Que remarquez-vous ?

3. Les bridges sont visibles via la commande `brctl show.`
   - Que remarquez-vous vis à vis des identifiants ?

4. Sur la machine hôte, afficher les routes (`route`, ou `ip route show`). Faire de même dans l'un des conteneurs.
   - Y-a-t-il des différences ? Si oui, lesquelles ?

Fournir un accès au conteneur depuis l'hôte

Lors du lancement du conteneur, le paramètre `-p` permet de spécifier quel(s) port(s) utiliser pour joindre le conteneur.
Ce paramètre accepte un valeur sous deux formes possibles :
- <containerPort>
- <hostPort>:<containerPort>

1. En vous basant sur la documentation docker, créer deux conteneurs **nginx,** le premier écoutant sur le port 80 du système hôte, le second sur le port 8080. Les deux conteneurs utiliseront le port interne 80.
2. Est-ce qu'à partir de l'hôte, ils sont accessibles ? (**localhost:8080** et **localhost:80**)
3. Est-ce qu'à partir des conteneurs, ils sont accessibles ? Si oui, à quelle adresse ? Utiliser l'image **alpine** et la commande **wget** pour les tests.

Tester le routage des réseaux (traceroute)

Créer deux conteneurs nommés **tr-1** et **tr-2** basés sur l'image **alpine** et les attacher à un réseau **interne** appartenant au sous-réseau **10.1.0.0/29**.

- Exécuter dans l'un d'eux (**tr-1**) la commande `traceroute tr-2`.
  Déduire du résultat l'interface utilisée ?
- Afficher les IPs du conteneur **tr-2**, et exécuter (toujours dans le conteneur **tr-1**) la commande `traceroute <ip>` (où l'IP est celle assignée par Docker, commençant en général par "172.17").
- Déconnecter le conteneur **tr-2** du réseau **interne** ; Créer un 3ème conteneur **tr-3**, le connecter au réseau interne, puis reconnecter **tr-2** au réseau interne ;
  Remarquez-vous des changements au niveau des IPs ?