docker-network-create

Créer un nouveau réseau

OPTIONS

–aux-address=map [ ] Adresses ipv4 ou ipv6 auxiliaires utilisée par le périphérique réseaux

-d, –driver=DRIVER Le pilote pour gérer le réseau ou l'overlay. Défaut : bridge

–gateway= [ ] passerelle IPv4 ou IPv6 pour le sous-réseaux maître

–internal Restreint l'accès externe au réseaux

–ip-range= [ ] Alloue une ip au conteneur dans une plage

–ipam-driver=default Pilote de gestion d'adresse ip

–ipam-opt=map [ ] Définis les options du pilote IPAM

-o, –opt=map [ ] Définis les options du pilote

–subnet= [ ] Sous-réseau au format CIDR qui représente un segment réseau

Le pilote peut être bridge ou overlay, qui sont des pilotes intégrés. Un pilote tier peut également être spécifié. Sans l'option –driver, la commande créé automatiquement un bridge. En installant Docker Engine, un bridge est installé automatiquement. Ce réseau correspond au bridge docker0 sur lequel le moteur s'appuie. En lançant un nouveau conteneur avec docker run, il est automatiquement connecté à ce bridge. On ne peut pas le supprimer, mais on peut en créer de nouveaux.

Les réseaux bridge sont des réseaux isolés sur une simple installation Engine. Pour créer un réseaux partagé avec plusieurs hôtes Docker, il faut créer un réseaux overlay. À la différence de bridge, overlay nécessite certaines conditions avant de pouvoir en créer un :

L'accès à un magasin de clé-valeur. Engine support Consul, Etcd, et Zookeeper. Un clustre d'hôte avec une connectivité à ce store. Un Engine configuré correctement sur chaque hôte du cluster. Le service docker supporte les options suivantes pour le réseaux Docker : **–cluster-store**, **–cluster-opt**, et **–cluster-advertise**.

C'est également une bonne idée, bien que non requis, d'installer Docker Swarn pour gérer le cluster. Swarn fournis une découverte sophistiquée et un gestionnaire de serveur qui peut assister l'implémentation.

Une fois les prérequis pour le réseau overlay préparé, il suffit de créer le réseau avec :
docker network create -d overlay my-multihost-network
Les noms de réseaux doivent être unique. Le service Docker tente d'identifier les conflicts de nommage mais ce n'est pas garantit.

Connecter des conteneurs

Au démarrage d'un conteneur utiliser –net pour se connecter à un réseaux. Cet exemple ajoute le conteneur busybox au réseau mynet :
docker run -itd –net=mynet busybox
Pour ajouter un conteneur à un réseau après que le conteneur soit démarré, utiliser **docker network connect**. Il est possible de connecter plusieurs conteneurs sur le même réseau. Une fois connectés, les conteneurs peuvent communiquer en eux.

Spécifier des options avancée

En créant un réseaux, Engine créé un sous-réseau sous le réseau par défaut. Ce sous-réseaux n'est pas une sous-division d'un réseau existant. C'est purement dans un but d'adressage ip. Il est possible de changer et de spécifier des valeurs de sous-réseau directement en utilisant l'option –subnet. sur un bridge, on peut seulement créer un simple sous-réseau :

docker network create -d bridge –subnet=192.168.0.0/16 br0

Additionnellement, on peut également spécifier les options **–gateway**, **–ip-range**, et **–aux-address**

network create –driver=bridge –subnet=172.28.0.0/16 –ip-range=172.28.5.0/24 –gateway=172.28.5.254 br0

Si –gateway est omis, Engine en séléctionne un dans le pool. Pour les réseaux overlay et pour les plugins réseaux qui le supporte on peut créer plusieurs sous-réseauxs

docker network create -d overlay –subnet=192.168.0.0/16 –subnet=192.170.0.0/16 –gateway=192.168.0.100 –gateway=192.170.0.100 –ip-range=192.168.1.0/24 –aux-address a=192.168.1.5 –aux-address b=192.168.1.6 –aux-address a=192.170.1.5 –aux-address b=192.170.1.6 my-multihost-network

Mode réseau interne

Par défaut, en connectant un réseau à un réseau overlay, Docker connecte également un réseau bridge pour fournir la connctivité externe. Pour créer un réseau overlay isolé, spécifier l'option **–internal**