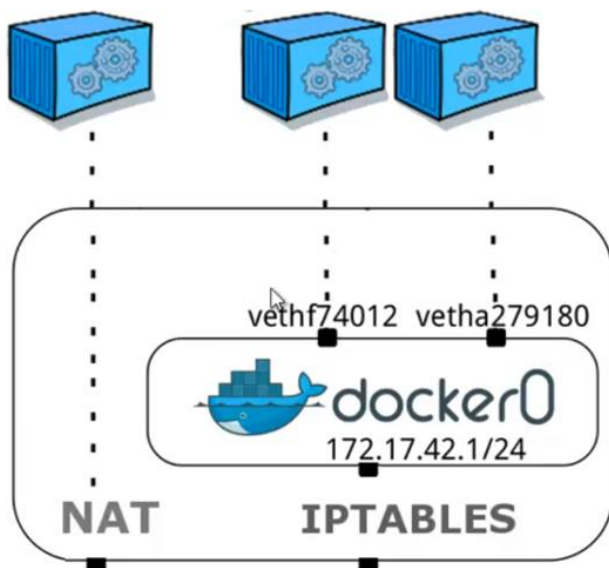


Le réseau sous Docker



- Créez un conteneur nommé network basé sur l'image busybox
- Repérez son adresse IP et notez là
- A-t-il accès à Internet ?
- Sortez de conteneur en le laissant actif
- Créez un second conteneur nommé network2 basé sur la même image
- Repérez son adresse IP
- Que remarquez vous ?
- Que peut-on en conclure ?
- Vérifiez les adresses IP de votre machine hôte afin d'observer nos remarques précédentes

A chaque fois qu'un conteneur sera créé, une IP lui sera automatiquement attribuée par un DHCP basé sur un bridge. Le réseau sera similaire à un réseau NAT

- Afin de vérifier ceci, tapez la commande suivante :

```
iptables -t nat -L
```

- Observez le MASQUERADE

C'est ce que vous avez réalisé précédemment avec l'option -p de vos serveurs web, vous avez effectué du NAT entre le port 80 de votre carte physique et le port 8080 du conteneur

Visualisez la liste des réseaux avec la commande

```
docker network ls
```

Nous observons donc 3 réseaux, le réseau bridge, celui qui est utilisé par défaut (celui des questions précédentes), le réseau null, qui en réalité n'est connecté nulle part, et le réseau host qui sera associé à l'adresse ip de la machine hôte.

- Créez un conteneur nommé host relié au réseau host basé sur busybox
- Visualisez son IP
- Concluez
- Créez un nouveau réseau nommé reseau1
- Effectuez une inspection de ce réseau et visualisez son adresse
- Effectuez un ip a sur votre machine hôte, que remarquez vous ?
- Créez un nouveau réseau nommé reseau2
- Effectuez un ip a sur votre machine hôte
- Créez un conteneur nommé ctn_reseau1 connecté au reseau1 basé sur busybox
- Repérez son IP
- A-t-il accès à Internet ?
- Sortez du conteneur sans le stopper
- Créez un conteneur nommé ctn_reseau2 connecté au reseau2 basé sur busybox
- Repérez son IP
- A-t-il accès à Internet ?

➤ Peut-il ping le conteneur précédent ?

➤ Concluez