

# Utilisation container

## Création de container

### Création

Lancer un conteneur basique

```
docker container run --publish 80:80 nginx  
ou  
docker container run -p 80:80 nginx
```

Lancer un conteneur mais en arrière plan dans cette configuration il aura un nom aléatoire

```
docker container run --detach --publish 80:80 nginx  
ou  
docker container run -d -p 80:80 nginx
```

## Afficher existant

Voir tous les container même stoppés

```
docker container ls -a
```

Commande identique

```
docker ps
```

Pour choisir le nom du container

```
docker container run --detach --name nginx_web --publish 80:80 nginx  
ou
```

```
docker container run -d --name nginx_web -p 80:80 nginx
```

# Arrêter un container

## Stopper

Pour stopper un container il faut son nom ou son ID

```
docker container stop 'NOM/ID'
```

Idem avec

```
docker stop 'NOM/ID'
```

## Supprimer

Pour supprimer le ou les container qui sont arrêtés mais toujours présent en fonds  
Il est possible d'en supprimer plusieurs en ajoutant tous les noms a la suite les un des autres.

```
docker container rm 'NOM/ID'
```

Idem avec

```
docker rm 'NOM/ID'
```

Il n'est normalement pas possible de supprimer un container en cours d'exécution sauf en le forçant avec l'ajout suivant

```
docker rm -f 'NOM/ID'
```

# Utilisation des container

## Regarder les logs

Pour regarder les logs

```
docker container logs 'NOM/ID'
```

Idem avec

```
docker logs 'NOM/ID'
```

## Processus

Voir les processus en cours dans un container docker

```
docker container top 'NOM/ID'
```

Idem avec

```
docker top 'NOM/ID'
```

## Informations image

```
docker container inspect 'NOM/ID'
```

Idem avec

```
docker inspect 'NOM/ID'
```

Il est possible de formater le résultats pour n'avoir que certains résultats.  
En modifiant la valeur entre crochet pour correspondre a la recherche souhaitée.

```
docker container inspect --format '{{.NetworkSettings.IPAddress}}' 'NOM/ID'
```

Il faudra faire attention a suivre le cheminement que l'on peut retrouver dans la commande inspect. Autre par exemple State.Status

```
daemon off;
},
'State': {
  'Status': 'running',
  'Running': true,
  'Paused': false,
  'Restarting': false.
```

# Utilisation des ressources par les containers

```
docker container stats 'NOM/ID'
```

Idem avec

```
docker stats 'NOM/ID'
```

## Intervenir dans le container

### Lancer un container et intervenir dedans

Pour lancer un container la commande sera identique mais il y aura quelques options en plus pour accéder a un terminal

```
docker container run -it --name webserver -p 80:80 nginx bash
```

-t permet de simuler un terminal

-i permet de de garder le terminal ouvert et recevoir l'output des commandes

bash correspond a la commande qui est lancée au lancement du terminal, il permet de créer un terminal

On entre directement dans la console du container

pour quitter le container il faut entrer `exit` mais cette commande éteint aussi le container

La raison pour laquelle le container s'éteint c'est que la commande pour faire fonctionner le container a été défini avec bash donc si l'on met fin au bash on met fin au container par contre s'il on execute une autre commande puis avec la syntaxe ci-dessous que l'on execute une commande bash supplémentaire le container restera allumé.

# Lancer une commande dans un container

```
docker container exec -it 'NOM/ID' 'commande'
```

ou

```
docker exec -it 'NOM/ID' 'commande'
```

la commande pour accéder a un terminal reste bash :

```
docker exec -it 'NOM/ID' bash
```

---

Revision #2

Created 23 July 2024 08:40:14 by maxime

Updated 23 July 2024 08:56:48 by maxime