Java dans Docker : bonnes pratiques

Speakers Charles Sabourdin (Freelance), Jean-Christophe Sirot (Docker)
Format : Tools in action
Date : 18 avril 2018

Dans ce Tools in Action, Charles et Jean-Christophe réalisent différents tests à partir de l'application Java disponible dans le repo GitHub jcsirot/resourcesmonger. Cette application est packagée sous forme d'un Fatjar et dispose d'un Dockerfile.

Test 1
L'application consomme toute la RAM de la machine, à savoir 256Mo.
Surprise : pas d'OutOfMemoryError (OOME) ni d'exception. Un simple message « Killed ».

Test 2
Les speakers augmentent la RAM à 512. Cette fois on tombe sur un OutOfMemoryError : Java Heap Space.

Test 3
Le Xmx est augmenté à 1024.
L'appli commence à ramer puis tombe en OOME.

Le –m 512M précise à Docker la quantité de mémoire disponible dans le conteneur. Lorsque le processus consomme plus de mémoire que donné, l'OS tue le processus.
Pourtant, l'application a réussi à utiliser 848Mo de Heap.
Explication : utilisation du swap qui permet de faire basculer la mémoire vive sur le disque. Lorsqu'il n'y a plus de place, l'OS commence à écrire sur le disque le contenu de la RAM. Il y'a des déplacements de données RAM <-> Disque dans tous les sens. D'où l'application qui rame.

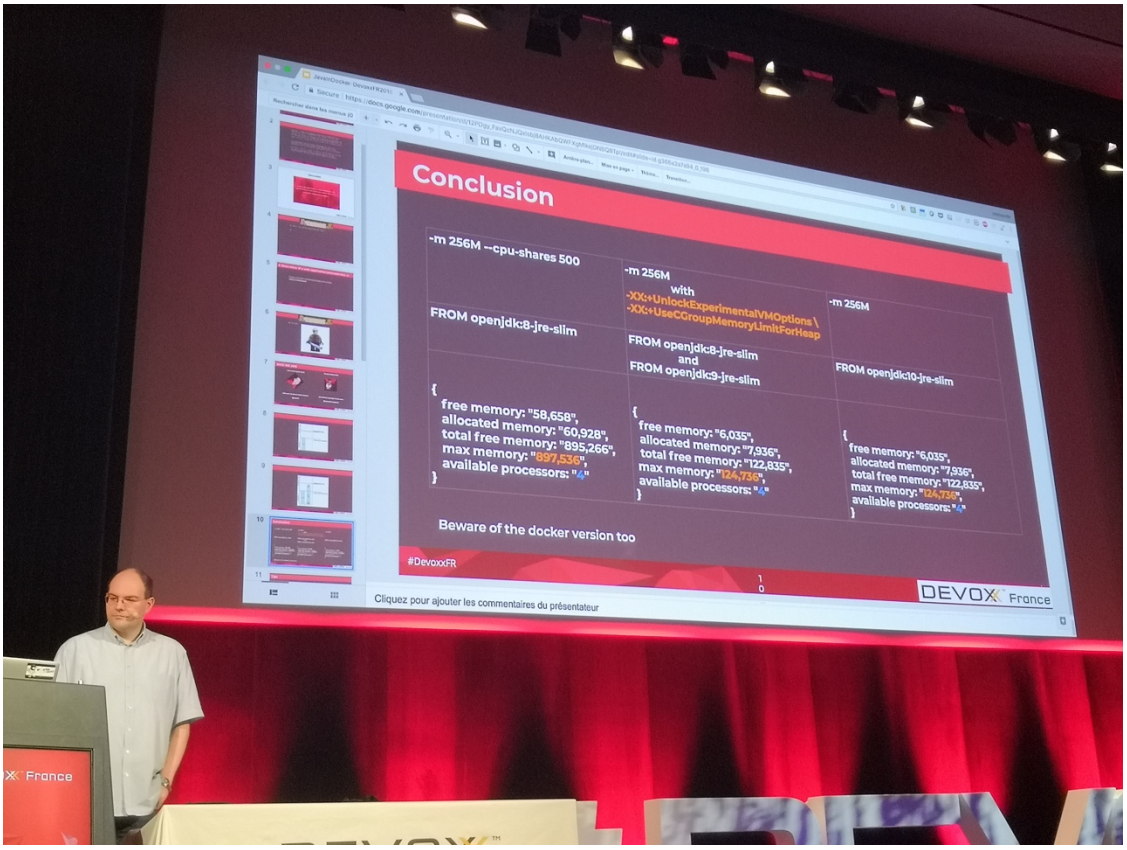
On pourrait forcer l'OS à ne plus utiliser le Swap.
Option Docker : --memory-swapiness=0
L'appli crashe plus vite, sans ramer. L'allocation du Heap est monté à 488Mo.

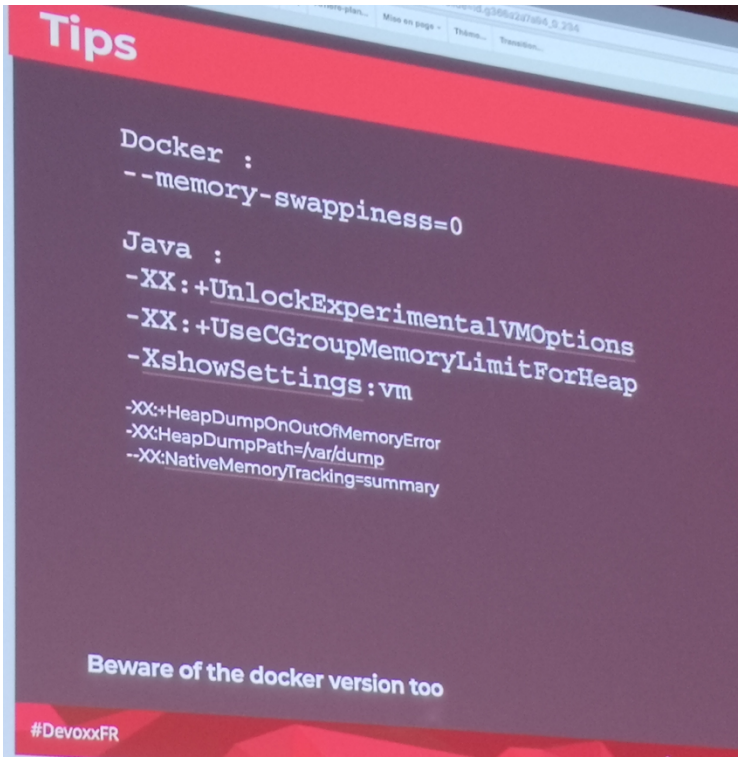
Comment affecter la bonne mémoire ?
Aligner –m et –Xmx requiert une formule complexe à établir.
Recommandation de Docker : utiliser Java 9 et non Java 8. Les développeurs de la JVM se sont posés la question.
Docker utilise le mécanisme des cgroup. Lorsqu'on démarre un conteneur, on peut fixer une limite de mémoire.

Depuis Java 9 et Java 8 update 131, des flags expérimentaux ont été ajoutés à la JVM : UnlockExperimentalVMOptions et UseCGroupMemoryLimitForHeap.
La JVM utilise ¼ de la mémoire disponible (si Xmx non spécifié).
En Java 10, ces options ne sont plus expérimentales et correspondent même au comportement par défaut (si pas de Xmx). Recommandation : enlever le Xmx.

La version 10 de Java apporte une meilleure intégration dans les conteneurs Docker.

Slide de conclusion du Tools in action :





Questions ouvertes :

- Où mettre ces options de paramétrage mémoire : orchestrateur, Dockerfile ?
- Les outils de build : plugin maven fabricate ? utilisation de jlink pour avoir des versions de JVM plus petites