

Université de Montréal

Documentation sur les flags du Cryptomator

Par :

Insafe Kibboua

Soundous Boutarfaya

Bac en informatique Faculté Arts et Sciences

Travail présenté à Monsieur Benoit Baudry dans le cadre du cours :

3913 Qualité logiciel et métriques

Date : 14 novembre 2024

Cette documentation décrit les modifications apportées à notre action GitHub pour permettre l'exécution de l'application avec cinq flags JVM spécifiques. Ces modifications visent à optimiser la performance et la gestion de la mémoire de notre application Cryptomator. Le fichier YAML de l'action GitHub a été modifié pour inclure des options JVM spécifiques dans la variable `MAVEN_OPTS` :

```
- name: Build and Test
  run: |
    # Flags to test
    flags=(
      "-Xms512m"
      "-XX:+UseG1GC"
      "-XX:+PrintGCDetails"
      "-XX:+HeapDumpOnOutOfMemoryError"
      "-XX:+UseCompressedOops"
    )
```

Afin de permettre la journalisation des flags, nous avons ajouté cette boucle qui permettra l'impression.

```
for flag in "${flags[@]}; do
  echo "Using MAVEN_OPTS: $flag"
  export MAVEN_OPTS="$flag"
```

Le choix de chaque flag :

1. -Xms512m

- **Description:** Définit la taille initiale de la mémoire heap à 512 MB.
- **Type:** Configuration de mémoire.
- **Impact sur la qualité:** Assure que l'application dispose d'une quantité de mémoire suffisante dès le démarrage, réduisant le risque de problèmes d'allocation au début de l'exécution.
- **Impact sur la performance:** Évite les cycles de redimensionnement de la heap, ce qui améliore le temps de démarrage et la réactivité initiale de l'application.
- **Impact sur l'observabilité:** La mémoire allouée dès le départ permet une meilleure visualisation de l'état de l'application sous une charge normale.

2. -XX:+UseG1GC

- **Description:** Active le Garbage Collector G1 (Garbage-First) qui est optimisé pour des faibles temps de pause.
- **Type:** Garbage Collection.
- **Impact sur la qualité:** Diminue les pauses causées par le ramassage des ordures, ce qui améliore le temps de réponse de l'application.

- **Impact sur la performance:** Contribue à maintenir un bon niveau de performance pendant l'exécution, surtout pour les applications de grande taille et à faible latence.
- **Impact sur l'observabilité:** Permet de collecter des métriques détaillées sur le comportement du garbage collector, facilitant le diagnostic des performances.

3. -XX:+PrintGCDetails

- **Description:** Active la journalisation détaillée de chaque événement de garbage collection.
- **Type:** Journalisation.
- **Impact sur la qualité:** Aide à identifier les problèmes de mémoire et à diagnostiquer les goulets d'étranglement en matière de GC.
- **Impact sur la performance:** Peut introduire une légère surcharge de performance, mais cela est souvent compensé par les informations utiles qu'il apporte.
- **Impact sur l'observabilité:** Fournit des données précieuses sur les événements de garbage collection, permettant une surveillance proactive des performances de l'application.

4. -XX:+HeapDumpOnOutOfMemoryError

- **Description:** Génère un heap dump lorsque l'application rencontre une erreur d'OutOfMemory.
- **Type:** Diagnostic.
- **Impact sur la qualité:** Permet d'identifier les problèmes de mémoire en offrant une vue instantanée de l'état de la mémoire au moment de l'erreur.
- **Impact sur la performance:** Peut causer une pause lorsque le dump est généré, mais cela est acceptable pour le débogage des problèmes critiques de mémoire.
- **Impact sur l'observabilité:** Offre une analyse approfondie de l'utilisation de la mémoire, facilitant les identifications des fuitages de mémoire et des objets qui consomment trop d'espace.

5. -XX:+UseCompressedOops

- **Description:** Active l'utilisation de pointeurs compressés (Ordinary Object Pointers) pour référencer des objets dans la mémoire heap.
- **Type:** Optimisation de mémoire.
- **Impact sur la qualité:** Réduit l'empreinte mémoire de l'application, ce qui peut aider à prévenir des erreurs d'OutOfMemoryError.
- **Impact sur la performance:** En utilisant des pointeurs compressés, cela permet de mieux utiliser la mémoire disponible, particulièrement bénéfique pour des applications fonctionnant sur des architectures 64 bits.
- **Impact sur l'observabilité:** Cela peut rendre la visualisation de l'utilisation de la mémoire plus claire, car moins d'espace est utilisé pour les pointeurs, ce qui permet de surveiller d'autres aspects de la mémoire plus efficacement.