

Dépannage des applications IOX qui ne peuvent pas démarrer ou qui s'arrêtent de manière inattendue

Contenu

[Introduction](#)

[Informations générales](#)

[Informations de base sur la terminaison d'application IOX](#)

[Empêcher les conteneurs IOX de s'arrêter lors de l'arrêt de l'application/de la cible](#)

Introduction

Ce document décrit comment dépanner ou déboguer des applications IOX qui rencontrent des problèmes lors de leur démarrage ou de leur arrêt inattendu.

Informations générales

Lorsque vous développez des applications IOX, la meilleure approche consiste à le développer sur une autre plate-forme et/ou sandbox. Une fois votre application IOX testée et prête, elle peut être empaquetée et déployée sur un périphérique compatible IOX. Dans certains cas, ce déploiement ne fonctionne pas comme prévu, l'application peut s'arrêter de manière inattendue ou même ne pas démarrer.

Le comportement par défaut pour et IOX VM/container/application est de s'arrêter dès que la commande cible est terminée. Cela rend difficile le dépannage si quelque chose d'inattendu se produit alors que toutes les informations non persistantes disparaîtront. Une autre conséquence est qu'il faut beaucoup de temps pour jouer/expérimenter avec les fonctionnalités des applications IOX. Pour démarrer une autre application/cible/commande/script, vous devez au moins modifier le package.yaml, créer le nouveau package, désactiver votre application actuelle, mettre à niveau avec le nouveau package, réactiver l'application et le démarrer.

Informations de base sur la terminaison d'application IOX

Local Manager et le client IOX ne fournissent pas beaucoup d'informations sur la raison pour laquelle les applications/conteneurs/machines virtuelles se sont arrêtés. Heureusement, le **watchDog.log** suit ceci et vous fournit également le dernier code de sortie/retour de l'application. Bien que cela ne vous aide pas toujours, dans de nombreux cas, cela vous conduit à la raison que vous recherchez.

Pour récupérer le fichier **watchDog.log** avec le client IOX :

```
[jedepuydb ~]$ ioxclient app logs tail iox_docker_test watchDog.log 10
Currently active profile : default
Command Name: application-logs-tail
```

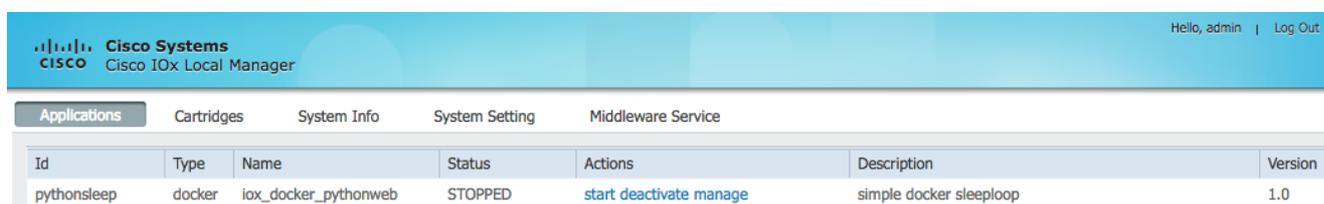
```

App/Service : iox_docker_test, Logfile : watchDog.log, viewing last 10 lines
APP END TIME:1498207460
Time taken by App : 0 minutes and 0 seconds.
Got the ip address - 10.197.215.227 for interface eth0
All interfaces got the ips
APP START TIME:1498207536
App iox_docker_test started with PID : 11
Monitoring this process now
App iox_docker_test completed with exit code: 127
APP END TIME:1498207536
Time taken by App : 0 minutes and 0 seconds.

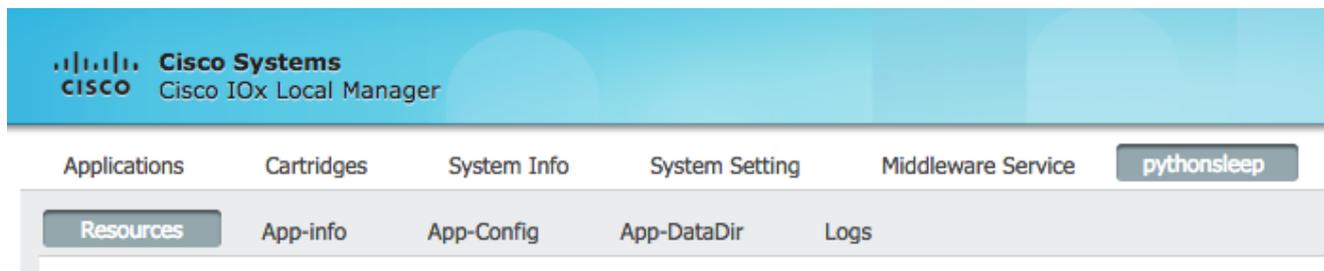
```

Pour récupérer le journal à l'aide du Gestionnaire local :

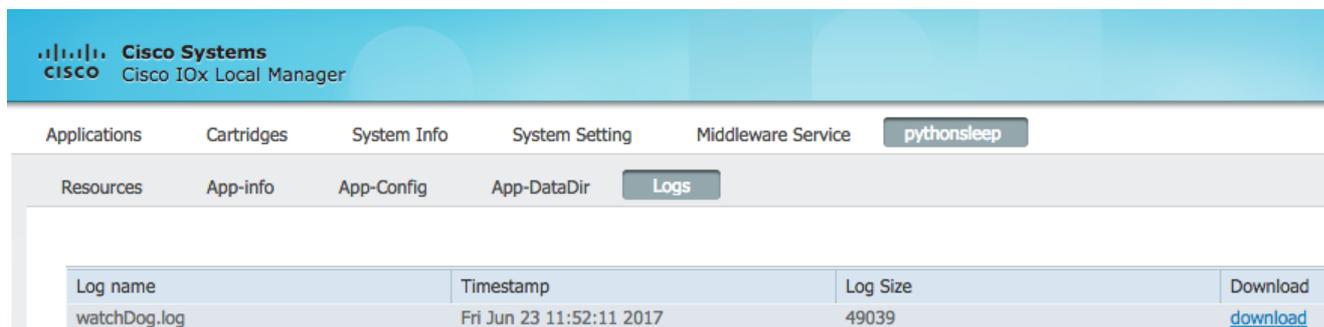
1. Connectez-vous à Local Manager.
2. Comme l'illustre l'image, cliquez sur **démarrer désactiver gérer** pour l'application concernée



3. Sélectionnez l'onglet **Journaux**, comme indiqué dans l'image :



4. Télécharger le **fichier watchDog.log**



Par exemple, cette application se termine de manière inattendue et vous le voyez dans le **fichier watchDog.log**

```

APP START TIME:1498207536
App iox_docker_test started with PID : 11
Monitoring this process now
App iox_docker_test completed with exit code: 127
APP END TIME:1498207536

```

Comme vous pouvez le voir dans l'extrait de journal ci-dessus, le code de sortie était 127. Il s'agit

d'un code de retour **réservé** et cela signifie : **commande introuvable**, qui indique qu'une commande défectueuse est spécifiée en tant que cible ou que le script qui démarre tente d'appeler une commande défectueuse.

Codes de sortie réservés les plus courants :

RC	Signification	commentaire
1	Rechercher les erreurs générales	Erreurs diverses, telles que « diviser par zéro » et autres opérations non autorisées
2	Utilisation abusive de bâtiments à coque	Mot clé ou commande manquant ou problème d'autorisation (et code de retour diff sur une comparaison de fichiers binaires échouée).
126	Impossible d'exécuter la commande	Le problème d'autorisation ou la commande n'est pas un exécutable
127	Commande introuvable	Problème possible avec \$PATH ou une faute de frappe
128	Argument de sortie non valide (RC>255)	exit prend uniquement des nombres entiers args dans la plage 0 - 255
+		
128	Signal d'erreur fatal « n »	le code de sortie renvoie 137 (128 + 9) -> le signal d'erreur de l'application n
130	Terminé par Ctrl+c	Control-C est le signal d'erreur fatal 2 (130 = 128 + 2, voir ci-dessus)
255	Statut de sortie hors limites (RC>255)	exit prend uniquement des nombres entiers args dans la plage 0 - 255

Pour plus d'informations à ce sujet, visitez le site : <http://tldp.org/LDP/abs/html/exitcodes.html>

Empêcher les conteneurs IOX de s'arrêter lors de l'arrêt de l'application/de la cible

La rubrique ci-dessus fournit des informations sur le dépannage d'une application défailante, mais elle n'empêche pas l'application IOX de s'arrêter. Cela signifie que les informations les plus utiles en matière de dépannage ont disparu, car toutes les données non persistantes n'existent plus.

Comme mentionné ci-dessus, un autre cas d'utilisation est de jouer avec les fonctionnalités de l'application IOX démarrée ou d'être flexible avec des commandes et des arguments.

Pour empêcher les applications IOX de se terminer à la fin de l'application, vous pouvez passer le **-debug on** à la commande activée :

```
[jedepuyd@db ~]$ ioxclient app activate -debug on testdebug
Currently active profile : default
Command Name: application-activate
App testdebug is Activated
[jedepuyd@db ~]$ ioxclient app start testdebug
Currently active profile : default
Command Name: application-start
App testdebug is Started
[jedepuyd@db ~]$ ioxclient app console testdebug
Currently active profile : default
Command Name: application-console
Console setup is complete..
Running command : [ssh -p 2222 -i testdebug.pem appconsole@10.48.43.197]
/ #
```

Dans l'exemple ci-dessus, après avoir activé et démarré avec l'indicateur **-debug on**, vous pouvez

accéder au conteneur même si l'application est terminée. Vous pouvez lancer d'autres commandes ici et expérimenter librement l'application dans l'environnement où votre application s'exécute. Cela permet de gagner beaucoup de temps en résolvant les problèmes d'application ou en obtenant la cible et les arguments appropriés.