

Fehlerbehebung bei IOX-Anwendungen, die nicht gestartet werden kann oder unerwartet beendet wird

Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[Grundlegende Informationen zur IOX-Anwendungsbeendigung](#)

[Verhindern, dass IOX-Container die Beendigung von Anwendungen/Zielterminierungen beenden](#)

Einführung

In diesem Dokument wird beschrieben, wie Sie IOX-Anwendungen, bei denen Probleme beim Einstieg auftreten oder unerwartet beendet werden, beheben oder debuggen können.

Hintergrundinformationen

Wenn Sie IOX-Anwendungen entwickeln, ist es am besten, diese auf einer anderen Plattform und/oder Sandbox zu entwickeln. Sobald die gewünschte IOX-Anwendung getestet und bereit ist, kann sie auf einem IOX-fähigen Gerät verpackt und bereitgestellt werden. In einigen Fällen funktioniert diese Bereitstellung nicht wie erwartet, die Anwendung kann unerwartet beendet werden oder sogar nicht starten.

Das Standardverhalten für und IOX VM/Container/Anwendung wird beendet, sobald der Zielbefehl beendet ist. Dies erschwert die Fehlerbehebung, wenn etwas Unerwartetes passiert, da alle nicht persistenten Informationen verschwinden. Eine weitere Folge ist, dass es sehr zeitaufwendig ist, wenn Sie die Funktionen von IOX-Anwendungen wiedergeben bzw. testen möchten. Wenn Sie eine andere App/ein anderes Ziel/einen anderen Befehl/ein anderes Skript starten, müssen Sie zumindest die Datei `package.yaml` ändern, das neue Paket erstellen, Ihre aktuelle Anwendung deaktivieren, mit dem neuen Paket aktualisieren, die Anwendung erneut aktivieren und starten.

Grundlegende Informationen zur IOX-Anwendungsbeendigung

Local Manager und IOX-Client liefern nicht viele Informationen darüber, warum Anwendungen/Container/VMs gestoppt wurden. Glücklicherweise verfolgt `watchDog.log` dies und stellt Ihnen den letzten Exit/Return-Code der Anwendung zur Verfügung. Auch wenn Ihnen das nicht immer hilft, führt es in vielen Fällen dazu, dass Sie suchen.

So rufen Sie das `watchDog.log` mit dem IOX-Client ab:

```
[jedepuyd@db ~]$ 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
```

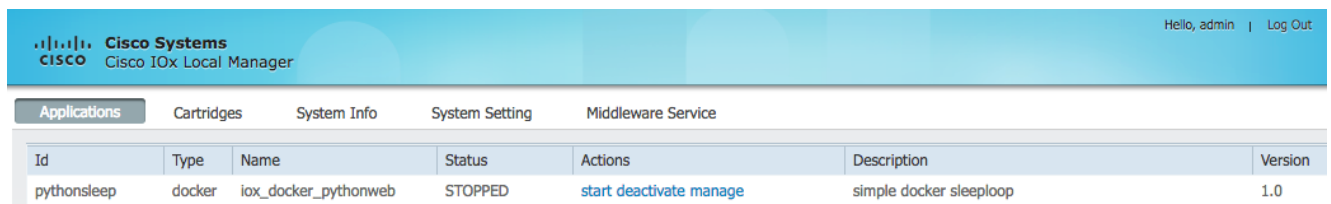
```

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.

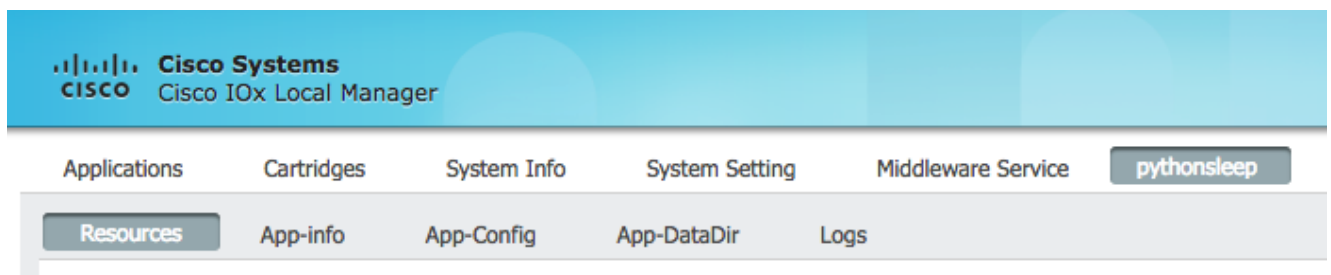
```

So rufen Sie das Protokoll mit dem lokalen Manager ab:

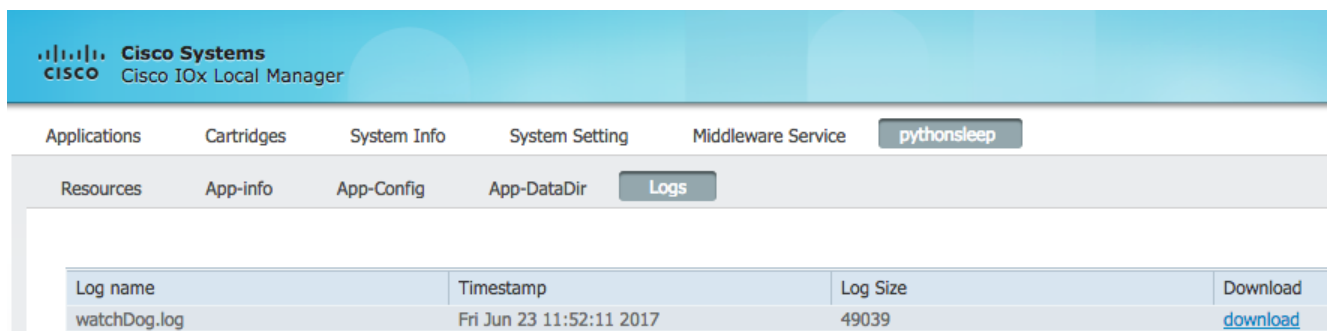
1. Melden Sie sich bei Local Manager an.
2. Wie im Bild gezeigt, klicken Sie auf **Start Deaktivieren verwalten** für die entsprechende Anwendung



3. Wählen Sie die Registerkarte **Protokolle**, wie im Bild gezeigt:



4. Laden Sie **watchDog.log** herunter



Beispielsweise wird diese Anwendung unerwartet beendet, und Sie sehen dies in **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

```

Wie Sie im obigen Protokollausschnitt sehen können, war der Exitcode 127. Dies ist ein **reservierter** Rückgabecode und bedeutet: **nicht gefunden**, was darauf hinweist, dass ein

fehlerhafter Befehl als Ziel angegeben wird oder das Skript, das gestartet wird, versucht, einen fehlerhaften Befehl aufzurufen.

Häufigste reservierte Exitcodes:

RC	Bedeutung	Kommentar
1	Catchall für allgemeine Fehler	Verschiedene Fehler wie "durch Null teilen" und andere unzulässige Vorgänge
2	Missbrauch von Shell Builtins	Fehlendes Schlüsselwort oder fehlender Befehl oder Berechtigungsproblem (und fehlender Rückgabecode bei einem fehlgeschlagenen Binärdateivergleich).
126	Befehl konnte nicht ausgeführt werden	Berechtigungsproblem oder Befehl ist keine ausführbare Datei
127	Befehl nicht gefunden	Mögliches Problem mit \$PATH oder einem Tippfehler
128	Ungültiges Argument zum Beenden (RC>255)	exit nimmt nur ganzzahlige Zeichen im Bereich 0-255 ein.
128+n	Schwerwiegendes Fehlersignal "n"	Exitcode gibt 137 (128 + 9) zurück -> Anwendungsfehler-Signal war 9
130	Beendet durch Strg+c	Control-C ist ein tödliches Fehlersignal 2, (130 = 128 + 2, siehe oben)
255	Exit-Status außerhalb des Bereichs (RC>255)	exit nimmt nur ganzzahlige Zeichen im Bereich 0-255 ein.

Weitere Informationen hierzu finden Sie hier: <http://tldp.org/LDP/abs/html/exitcodes.html>

Verhindern, dass IOX-Container die Beendigung von Anwendungen/Zielterminierungen beenden

Das obige Thema enthält Informationen zum Beheben von Fehlern bei Anwendungen, verhindert jedoch nicht, dass die IOX-Anwendung beendet wird. Dies bedeutet, dass die wahrscheinlich wertvollsten Informationen zur Fehlerbehebung verschwunden sind, da alle nicht persistenten Daten nicht mehr vorhanden sind.

Wie bereits erwähnt, ist ein weiterer Anwendungsfall die Nutzung der Funktionen der startenden IOX-Anwendung oder die Flexibilität bei Befehlen und Argumenten.

Um zu verhindern, dass IOX-Anwendungen am Anwendungsende beendet werden, können Sie das **-Debuggen** an den Befehl `enable` übergeben:

```
[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]
/ #
```

Im obigen Beispiel können Sie nach dem Aktivieren und Beginnen des -debug on-Flags auf den Container zugreifen, selbst wenn die Anwendung beendet wird. Sie können hier andere Befehle starten und frei mit der Anwendung in der Umgebung experimentieren, in der die Anwendung ausgeführt wird. Dies spart viel Zeit bei der Behebung von Anwendungsproblemen oder beim Einstellen der richtigen Ziele und Argumente.