

Pesquise defeitos aplicativos IOX que é incapaz de começar ou pare inesperadamente

Índice

[Introdução](#)

[Informações de Apoio](#)

[Informação básica na terminação do aplicativo IOX](#)

[Impedindo que IOX os recipientes pare no aplicativo/visam a terminação](#)

Introdução

Este documento descreve como pesquisar defeitos ou debugar aplicativos IOX que encontra a edição na obtenção começado ou para-o inesperadamente.

Informações de Apoio

Quando você desenvolve aplicativos IOX, a melhor aproximação é desenvolvê-lo em uma outra plataforma e/ou caixa de areia. Uma vez que seu aplicativo desejado IOX é testado e se apronta, pode ser empacotado e distribuído em um dispositivo da IOX-possibilidade. Em alguns casos, esse desenvolvimento não trabalha como esperado, o aplicativo pôde parar inesperadamente ou pode nem sequer começar.

O comportamento padrão para e o IOX VM/container/application são parar assim que o comando do alvo terminar. Isto faz difícil pesquisar defeitos se algo inesperado acontece porque toda a informação não-persistente estará desaparecida. Uma outra consequência é que é muito demorada em que você quer jogar/experiência com as capacidades de aplicativos IOX. Começar um outro app/alvo/comando/script exige-lo-ia mudar pelo menos o package.yaml, construir o pacote novo, desativar seu aplicativo atual, promovê-lo com o pacote novo, re-ativar o aplicativo e começá-lo.

Informação básica na terminação do aplicativo IOX

O gerente local e o cliente IOX não fornecem muita informação na razão pela qual os aplicativos/recipientes/VM pararam. Felizmente, **watchDog.log** mantém-se a par deste e igualmente fornece-se o a últimos saída/código de retorno do aplicativo. Embora isto não possa sempre o ajudar, em muitos casos condu-lo à razão que você está procurando.

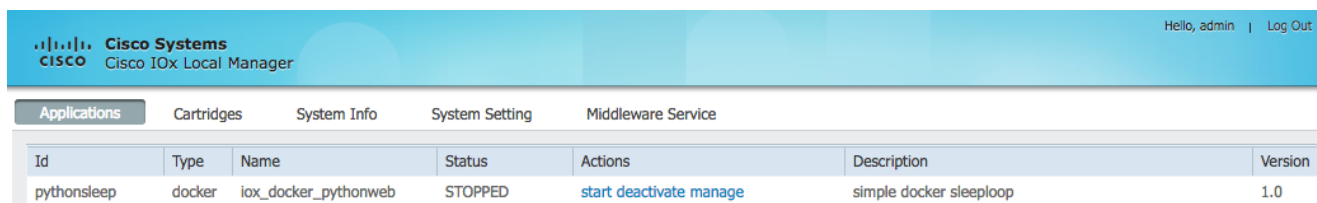
Para buscar **watchDog.log** com cliente IOX:

```
[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
```

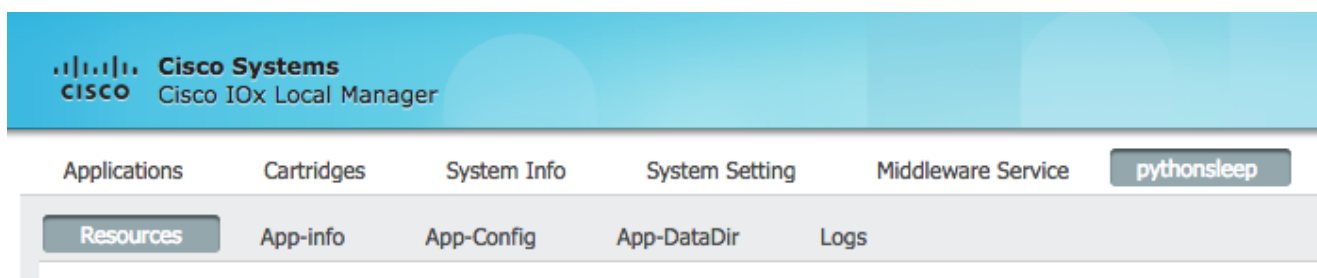
```
APP START TIME:1498207536
App iox_docker_test started with PID : 11
Monitoring this process now
App iox_docker_test completed with exit code: 127
APP END TIME:1498207536
Time taken by App : 0 minutes and 0 seconds.
```

Para buscar o log usando o gerente local:

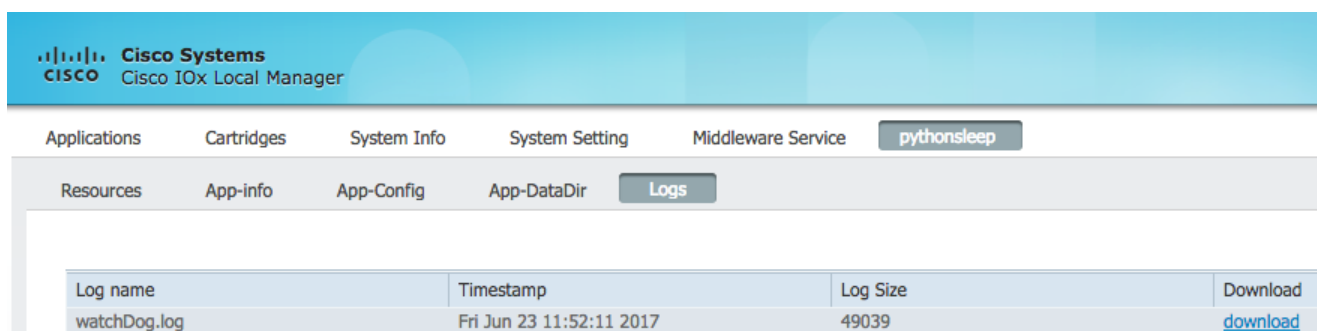
1. Início de uma sessão ao gerente local.
2. Segundo as indicações da imagem, clique sobre o **começo desativam controlam** para o aplicativo relevante



3. Selecione a aba dos **logs**, segundo as indicações da imagem:



4. Transfira o **watchDog.log**



Por exemplo, este aplicativo é terminado inesperadamente e você vê este no **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
```

Como você pode ver no snippet acima do log, o código de saída era 127. Este é um código de retorno **reservado** e significa: **o comando não encontrado**, que indica um comando defeituoso é especificado enquanto um alvo ou o script que seja tentativas começadas para chamar um comando defeituoso.

A maioria de códigos de saída reservados comuns:

RC	Significado	Comentário
1	Saco para erros gerais	Erros variados, tais como a "partilha por zero" e por outras operações inadmissíveis
2	Emprego errado de builtins do shell	Palavra-chave ou comando faltante, ou problema de permissão (e código retorno do diff em uma comparação falhada do arquivo binário).
126	O comando não podia ser executado	O problema de permissão ou o comando não são um executável
127	Comando não encontrado	Problema possível com \$PATH ou um erro tipográfico
128	Argumento inválido a retirar (RC>255)	retire args do inteiro das tomadas somente na escala 0 - 255
128+n	Sinal "n" do erro fatal	o código de saída retorna 137 (128 + 9) - > sinal do erro do app era 9
130	Terminado por Ctrl+c	O Control-c é o sinal 2 do erro fatal, (130 = 128 + 2, veem acima)
255	Status de saída fora da escala (RC>255)	retire args do inteiro das tomadas somente na escala 0 - 255

Mais informação nesta pode ser encontrada aqui: <http://tldp.org/LDP/abs/html/exitcodes.html>

Impedindo que os recipientes IOX parem na terminação do aplicativo/alvo

O assunto acima fornece a informação em como pesquisar defeitos um aplicativo de falha mas não impede que o aplicativo IOX pare. Isto significa que muito provavelmente relativo à informação valioso à pesquisa de defeitos desapareceu enquanto todos os dados não-persistentes já não existem.

Como mencionado acima, um outro caso do uso é jogar com as capacidades do IOX-aplicativo começado ou ser flexível com comandos e argumentos.

Para impedir que os aplicativos IOX terminem na extremidade do aplicativo, você pode passar -**debug sobre ao** comando da ativação:

```
[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]
/ #
```

No exemplo acima, após a ativação e começar com - debugar na bandeira, você pode alcançar o recipiente mesmo se o aplicativo é terminado. Você pode lançar outros comandos aqui e pode livremente experimentar com o aplicativo no ambiente aonde seu aplicativo é executado. Isto salvar muito tempo em resolver edições do aplicativo ou em obter o alvo e os argumentos diretos ajustados.