

Solucionar problemas de aplicativos IOX que não podem ser iniciados ou interrompidos inesperadamente

Contents

[Introduction](#)

[Informações de Apoio](#)

[Informações básicas sobre a terminação de aplicativos IOX](#)

[Impedindo que os contêineres de IOX parem com a terminação de aplicativos/alvos](#)

Introduction

Este documento descreve como solucionar problemas ou depurar aplicativos IOX que encontram problemas para iniciar ou parar inesperadamente.

Informações de Apoio

Ao desenvolver aplicativos IOX, a melhor abordagem é desenvolvê-los em outra plataforma e/ou sandbox. Depois que o aplicativo IOX desejado for testado e estiver pronto, ele poderá ser empacotado e implantado em um dispositivo habilitado para IOX. Em alguns casos, se a implantação não funcionar como esperado, o aplicativo pode parar inesperadamente ou nem mesmo iniciar.

O comportamento padrão para a VM/contêiner/aplicação IOX e IOX é parar assim que o comando de destino terminar. Isso dificulta a solução de problemas se algo inesperado acontecer, pois todas as informações não persistentes serão desaparecidas. Outra consequência é que é muito demorado quando você quer jogar/experimentar com os recursos dos aplicativos IOX. Iniciar outro aplicativo/alvo/comando/script exigiria que pelo menos altere o pacote.yaml, crie o novo pacote, desative o aplicativo atual, atualize com o novo pacote, reative o aplicativo e inicie-o.

Informações básicas sobre a terminação de aplicativos IOX

O Local Manager e o cliente IOX não fornecem muitas informações sobre o motivo pelo qual os aplicativos/contêineres/VMs pararam. Felizmente, o **watchDog.log** controla isso e também fornece o último código de saída/retorno do aplicativo. Embora isso nem sempre o ajude, em muitos casos, o motivo pelo qual você está procurando.

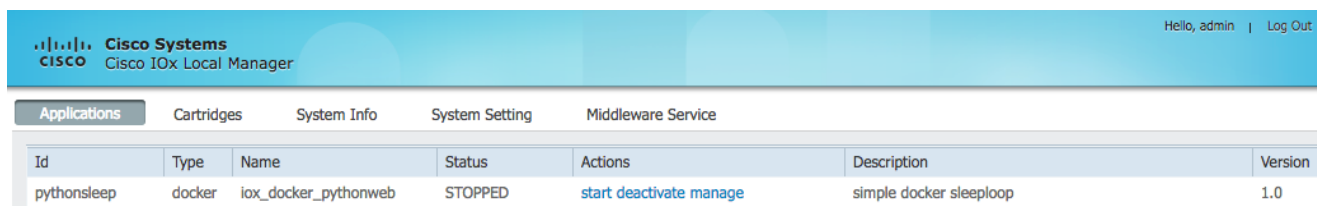
Para buscar o **watchDog.log** com o cliente IOX:

```
[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
```

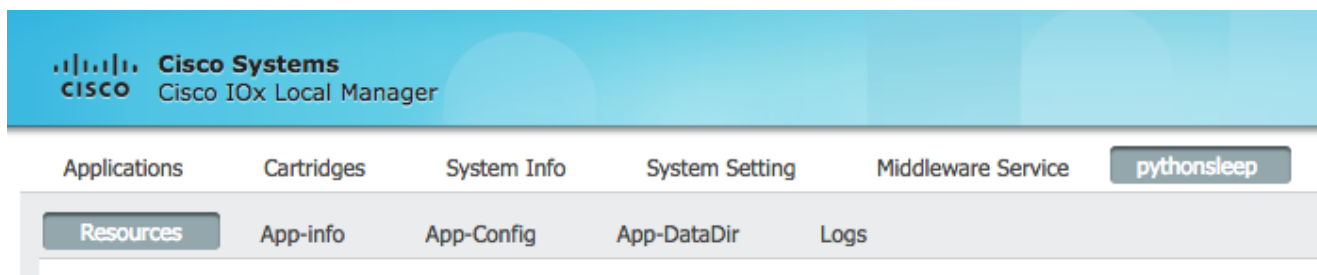
```
Time taken by App : 0 minutes and 0 seconds.
Got the ip address - 10.197.215.227 for interface eth0
All interfaces got the ips
APP START TIME:1498207536
App iox_docker_test started with PID : 11
Monitoring this process now
App iox_docker_test completed with exit code: 127
APP END TIME:1498207536
Time taken by App : 0 minutes and 0 seconds.
```

Para buscar o log usando o Gerenciador local:

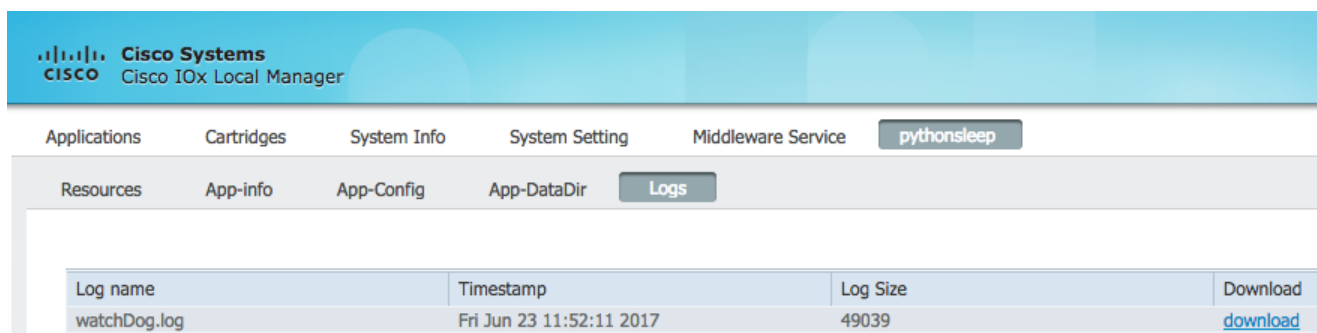
1. Faça login no Local Manager.
2. Como mostrado na imagem, clique em **Iniciar Desativar Gerenciar** para o aplicativo relevante



3. Selecione a guia **Logs**, conforme mostrado na imagem:



4. Baixe o **watchDog.log**



Por exemplo, esta aplicação é terminada inesperadamente e é apresentada 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 trecho de log acima, o código de saída foi 127. Este é um código de retorno **reservado** e significa: **comando não encontrado**, que indica que um comando defeituoso é especificado como um destino ou o script iniciado tenta chamar um comando defeituoso.

Códigos de saída reservados mais comuns:

RC	Significado	Comentário
1	Captura de erros gerais	Erros diversos, tais como "dividir por zero" e outras operações não autorizadas
2	Utilização abusiva de construções de conchas	Palavra-chave ou comando ausente ou problema de permissão (e código de retorno diff em uma comparação de arquivo binário com falha).
126	Não foi possível executar o comando	O problema de permissão ou comando não é executável
127	Comando não encontrado	Possível problema com \$PATH ou um erro de digitação
128	Argumento inválido para sair (RC>255)	exit aceita somente números inteiros no intervalo 0 a 255
128+n	Sinal de erro fatal "n"	código de saída retorna 137 (128 + 9) -> sinal de erro de aplicativo 9
130	Terminado por Ctrl+c	Control-C é um sinal de erro fatal 2, (130 = 128 + 2, ver acima)
255	Status de saída fora do intervalo (RC>255)	exit aceita somente números inteiros no intervalo 0 a 255

Mais informações sobre este assunto podem ser encontradas aqui:

<http://tldp.org/LDP/abs/html/exitcodes.html>

Impedindo que os contêineres de IOX parem com a terminação de aplicativos/alvos

O tópico acima fornece informações sobre como solucionar problemas de um aplicativo com falha, mas não impede que o aplicativo IOX pare. Isso significa que informações valiosas mais prováveis relacionadas à solução de problemas desapareceram, pois todos os dados não persistentes não existem mais.

Como mencionado acima, outro caso de uso é brincar com os recursos do aplicativo IOX iniciado ou ser flexível com comandos e argumentos.

Para impedir que os aplicativos IOX terminem na extremidade do aplicativo, você pode passar o **-debug on** para o comando ativate:

```
[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, depois de ativar e iniciar com o **-debug on** flag, você pode acessar o contêiner mesmo que o aplicativo seja terminado. Você pode iniciar outros comandos aqui e pode experimentar livremente o aplicativo no ambiente em que o aplicativo é executado. Isso economiza muito tempo na solução de problemas de aplicativos ou na obtenção do alvo e dos argumentos certos.