

# Risoluzione dei problemi relativi alle applicazioni IOX che non è possibile avviare o che si arrestano in modo imprevisto

## Sommario

[Introduzione](#)

[Premesse](#)

[Informazioni di base sulla terminazione delle applicazioni IOX](#)

[Impedire l'arresto dei contenitori IOX alla terminazione dell'applicazione o della destinazione](#)

## Introduzione

In questo documento viene descritto come risolvere i problemi o eseguire il debug delle applicazioni IOX in cui si verificano problemi durante l'avvio o arresti imprevisti.

## Premesse

Quando si sviluppano applicazioni IOX, l'approccio migliore è quello di svilupparle su un'altra piattaforma e/o sandbox. Una volta testata e pronta, l'applicazione IOX desiderata può essere assemblata e distribuita su un dispositivo abilitato per IOX. In alcuni casi, la distribuzione non funziona come previsto, l'applicazione potrebbe arrestarsi in modo imprevisto o non avviarsi.

Per impostazione predefinita, il comportamento di e IOX VM/container/application viene interrotto non appena il comando target viene terminato. Ciò rende difficile la risoluzione dei problemi in caso di eventi imprevisti, in quanto tutte le informazioni non persistenti scompariranno. Un'altra conseguenza è che l'esecuzione/sperimentazione delle funzionalità delle applicazioni IOX richiede molto tempo. Per avviare un'altra app/destinazione/comando/script è necessario almeno modificare il pacchetto.yaml, compilare il nuovo pacchetto, disattivare l'applicazione corrente, eseguire l'aggiornamento con il nuovo pacchetto, riattivare l'applicazione e avviarla.

## Informazioni di base sulla terminazione delle applicazioni IOX

Local Manager e il client IOX non forniscono molte informazioni sul motivo per cui le applicazioni/contenitori/macchine virtuali sono state arrestate. Fortunatamente, il file **watchDog.log** tiene traccia di questo e fornisce anche l'ultimo codice di uscita/ritorno dell'applicazione. Anche se questo potrebbe non sempre aiutarti, in molti casi ti porta al motivo che stai cercando.

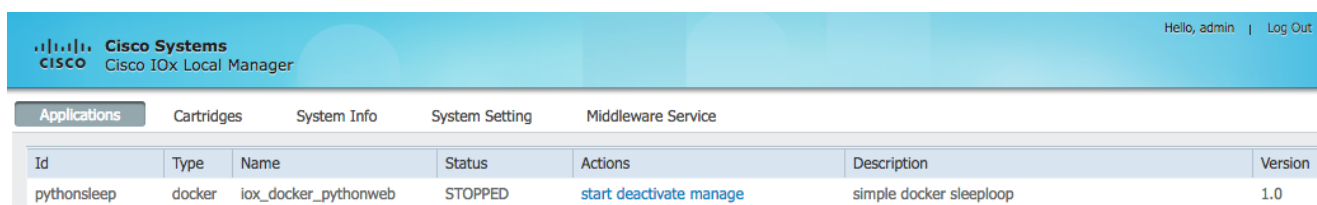
Per recuperare il file **watchDog.log** con il client IOX:

```
[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
```

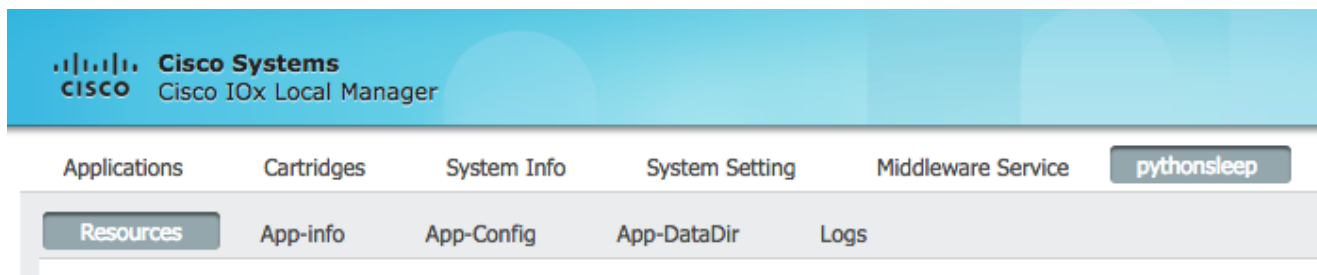
```
Time taken by App : 0 minutes and 0 seconds.
Got the ip address - 10.197.215.227 for interface eth0
All interfaces got the ips
APP START TIME:1498207536
App iox_docker_test started with PID : 11
Monitoring this process now
App iox_docker_test completed with exit code: 127
APP END TIME:1498207536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

Per recuperare il log mediante Gestione locale:

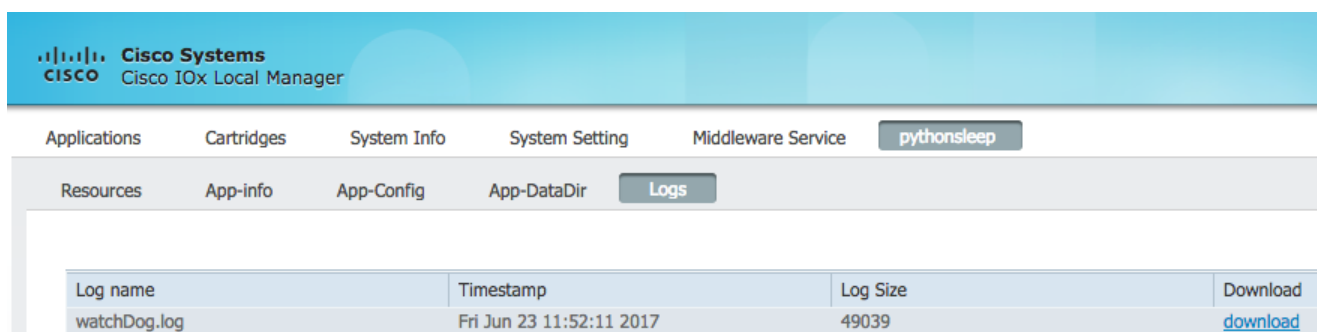
1. Accedere a Gestione locale.
2. Come mostrato nell'immagine, fare clic su **start deactivate manage** (disattiva la gestione per l'applicazione in questione)



3. Selezionare la scheda **Log**, come mostrato nell'immagine:



4. Scaricare il file **watchDog.log**



Questa applicazione, ad esempio, viene terminata in modo imprevisto e ciò viene visualizzato nel file **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
```

Come si può vedere nel frammento di log sopra riportato, il codice di uscita è 127. Questo è un codice restituito **riservato** e significa: **comando non trovato**, il che indica che un comando errato è

specificato come destinazione o che lo script avviato tenta di chiamare un comando errato.

Codici di uscita riservati più comuni:

RC	Significato	Commento
1	Catchall per errori generali	Errori vari, ad esempio "divisione per zero" e altre operazioni non consentite
2	Uso improprio di shell builtins	Parola chiave o comando mancante oppure problema di autorizzazione codice di restituzione diff in un confronto di file binari non riuscito).
126	Impossibile eseguire il comando	Il comando o il problema di autorizzazione non è eseguibile
127	Comando non trovato	Possibile problema con \$PATH o un errore di battitura
128	Argomento non valido per uscire (RC>255)	exit accetta solo argomenti di tipo integer compresi nell'intervallo da 0 a 255
128+n	Segnale di errore irreversibile "n"	il codice di uscita restituisce 128 + n -> il segnale di errore dell'errore è stato n
130	Terminato da Ctrl+c	Control-C è il segnale di errore irreversibile 2, (130 = 128 + 2, vedere sopra)
255	Stato uscita non compreso nell'intervallo (RC>255)	exit accetta solo argomenti di tipo integer compresi nell'intervallo da 0 a 255

Ulteriori informazioni al riguardo sono disponibili all'indirizzo:

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

## Impedire l'arresto dei contenitori IOX alla terminazione dell'applicazione o della destinazione

Nell'argomento precedente vengono fornite informazioni su come risolvere i problemi relativi a un'applicazione in errore, ma non viene impedito l'arresto dell'applicazione IOX. Ciò significa che molto probabilmente le informazioni importanti relative alla risoluzione dei problemi sono scomparse poiché tutti i dati non persistenti non esistono più.

Come accennato in precedenza, un altro caso di utilizzo è quello di giocare con le funzionalità dell'applicazione IOX avviata o essere flessibile con comandi e argomenti.

Per evitare che le applicazioni IOX vengano terminate all'estremità dell'applicazione, passare il comando **-debug on** al comando activate:

```
[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]
/ #
```

Nell'esempio precedente, dopo l'attivazione e l'avvio con il flag **-debug on**, è possibile accedere al

contenitore anche se l'applicazione viene terminata. È possibile avviare altri comandi e sperimentare liberamente l'applicazione nell'ambiente in cui viene eseguita. Ciò consente di risparmiare molto tempo nella risoluzione dei problemi dell'applicazione o nell'impostazione della destinazione e degli argomenti corretti.