

Устраните неполадки Приложений IOX, который Неспособен Запуститься или Неожиданно Остановы

Содержание

[Введение](#)

[Общие сведения](#)

[Основные сведения после завершения приложения IOX](#)

[Препятствование тому, чтобы контейнеры IOX остановились после завершения приложения/Цели](#)

Введение

Этот документ описывает, как устранить неполадки или отладить приложения IOX , который встречается с проблемой в начале работы или неожиданно останавливается.

Общие сведения

Когда вы developе IOX приложения, лучший подход - к developе он на другой платформе и/или песочнице. Как только ваше желаемое приложение IOX протестировано и готово, оно может быть упаковано и развернуто на IOX-разрешать устройстве. В некоторых случаях те развертывания не работают как ожидалось, приложение могло бы неожиданно остановиться или даже может не запуститься.

Поведение по умолчанию для и VM/контейнер/приложение IOX должны остановиться, как только завершилась целевая команда. Это мешает устранять неполадки, если что-то неожиданное происходит, поскольку исчезнет вся непостоянная информация. Другое последствие - то, что это является очень трудоемким, когда вы хотите играть/экспериментировать с возможностями приложений IOX. Начало другого приложения/цели/команды/сценария потребовало бы, чтобы вы, по крайней мере, изменили `package.yaml`, создали новый пакет, деактивировали ваше текущее приложение, обновление с новым пакетом, повторно активировали приложение и запустили его.

Основные сведения после завершения приложения IOX

Локальный Менеджер и клиент IOX не предоставляют большую информацию о причине, почему остановились приложения/контейнеры/VM. К счастью, **watchDog.log** отслеживает это и также предоставляет вам последний выход/код возврата приложения. Несмотря на то, что это не могло бы всегда помогать вам в большом количестве случаев, это ведет вас к причине, которую вы ищите.

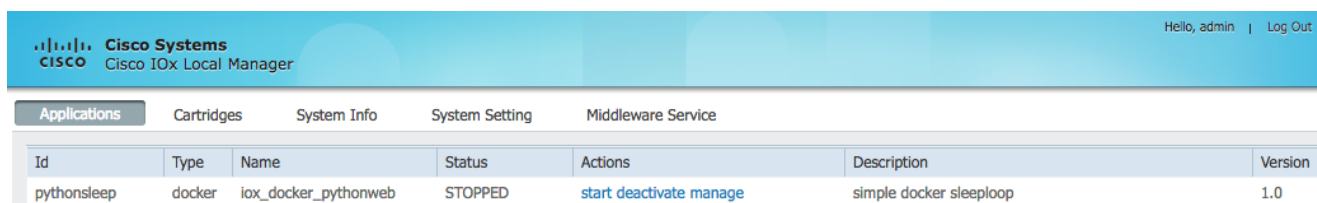
Выбирать **watchDog.log** с клиентом IOX:

```
[jedepuyd@db ~]$ ioxclient app logs tail iox_docker_test watchDog.log 10  
Currently active profile : default
```

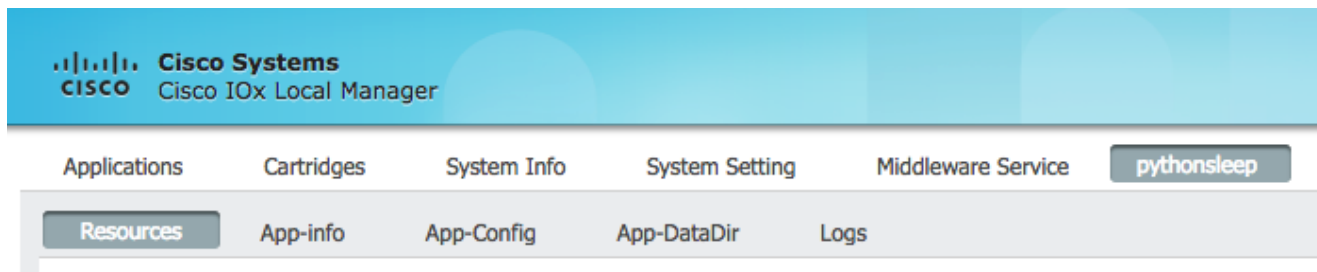
```
Command Name: application-logs-tail
App/Service : iox_docker_test, Logfile : watchDog.log, viewing last 10 lines
APP END TIME:1498207460
Time taken by App : 0 minutes and 0 seconds.
Got the ip address - 10.197.215.227 for interface eth0
All interfaces got the ips
APP START TIME:1498207536
App iox_docker_test 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.
```

Выбирать журнал с помощью Локального Менеджера:

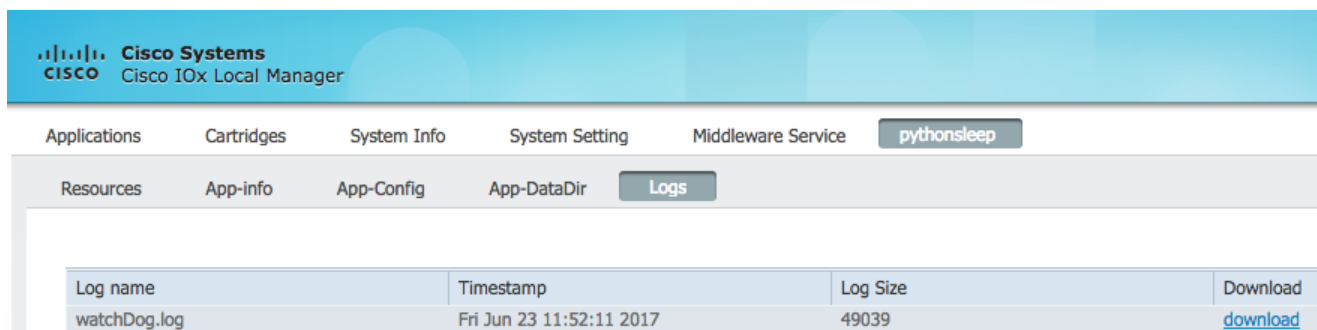
1. Войдите локальному менеджеру.
2. Как показано в образе, **нажмите кнопку Старт**, деактивируются, управляют для соответствующего приложения



3. Выберите вкладку **Logs**, как показано в образе:



4. Загрузите **watchDog.log**



Например, это приложение неожиданно завершено, и вы видите это в **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
```

Как вы можете видеть в вышеупомянутом регистрационном фрагменте, код выхода

равнялся 127. Это - **зарезервированный** код возврата, и это означает: **команда, не найденная**, который указывает на неисправную команду, задана как цель или сценарий, который запущен попытки вызвать неисправную команду.

Наиболее распространенные зарезервированные коды выхода:

RC	Значение	Комментарий
1	Вместилище для обычных ошибок	Различные ошибки, те, которые "делят на нулевые" и другие nepозволительные операции
2	Неправильное употребление оболочки builtins	Недостающее ключевое слово или команда или проблема разрешения (и diff код возврата на отказавшем сравнении двоичного файла).
126	Команда не могла быть выполнена	Проблема разрешения или команда не являются исполняемым файлом
127	Команда, не найденная	Возможная проблема с \$PATH или опечаткой
128	Недопустимый аргумент для выхода (RC > 255)	выход берет только целое число args в диапазоне 0 - 255
128+n	Сигнал "n" фатальной ошибки	код выхода возвращается 137 (128 + 9)->, сигнал ошибки приложен равнялся 9
130	Завершенный Ctrl+c	Ctrl-C является сигналом 2 фатальной ошибки, (130 = 128 + 2, посмотрите выше),
255	Статус выхода из диапазона (RC > 255)	выход берет только целое число args в диапазоне 0 - 255

Дополнительные сведения об этом могут быть найдены здесь: <http://tldp.org/LDP/abs/html/exitcodes.html>

Препятствование тому, чтобы контейнеры IOX остановились после завершения приложения/Цели

Вышеупомянутая тема предоставляет сведения о том, как устранить неполадки отказывающего приложения, но она не препятствует тому, чтобы остановилось приложение IOX. Это означает, что наиболее вероятная полезная информация, отнесенная к устранению проблем, исчезла, поскольку больше не существуют все непостоянные данные.

Как упомянуто выше, другой вариант использования должен играть с возможностями запущенного IOX-приложения или быть гибким с командами и аргументами.

Чтобы препятствовать тому, чтобы приложения IOX завершились на конце приложения, можно пройти - **отлаживают на активировать** команде:

```
[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]
```

```
/ #
```

В вышеупомянутом примере, после активации и начиная с - отлаживают на флаге, можно обратиться к контейнеру, даже если завершено приложение. Вы можете запустить другие команды здесь и можете свободно экспериментировать с приложением в среде где ваше выполнение приложения. Это экономит много времени в решении проблем приложения или получении правильной цели и набора аргументов.