

外部生命周期操作发生故障|码头工人

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[问题](#)

[验证](#)

[解决方案](#)

简介

当用户收到“无法访问JAR文件cliqr-repository-client-*-jar-with-dependencies.jar”错误时，本文描述如何解决问题。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档中的信息根据Cloud中心版本4.6和以上。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

问题

通常，当您设法实施与使用的一应用程序外部生命周期操作时，码头工人设法执行CCO启动的脚本。然而，因为不能读脚本由于的码头工人损坏，它失败。

错误消息：

```
2017-03-12 17:08:01,085 DEBUG service.LifecycleExternalServiceAction [nodeMessageTaskExecutor-12] - reading container script timeout value from gateway configuration 10m
2017-03-12 17:08:05,181 DEBUG container.GatewayContainerService [nodeMessageTaskExecutor-12] - 306: Output
Error: Unable to access jarfile cliqr-repository-client-*-jar-with-dependencies.jar^M
CLIQR_EXTERNAL_SERVICE_ERR_MSG_START^M
Failed to download files specified in environment variables^M
CLIQR_EXTERNAL_SERVICE_ERR_MSG_END^M
```

```
2017-03-12 17:08:05,182 ERROR service.LifecycleExternalServiceAction [nodeMessageTaskExecutor-12] -
java.lang.ArrayIndexOutOfBoundsException: 4
java.lang.ArrayIndexOutOfBoundsException: 4
    at
com.osmosix.gateway.container.ContainerWorkerBuilder.build(ContainerWorkerBuilder.java:31)
    at
com.osmosix.gateway.container.GatewayContainerService.execute(GatewayContainerService.java:94)
    at
com.osmosix.gateway.lifecycle.action.orchestration.service.LifecycleExternalServiceAction.executeContainerCommand(LifecycleExternalServiceAction.java:198)
```

注意：在GUI中，它只显示与Error 4。

要纠正此问题，您需要遵从这些步骤：

验证

从CCO/Docker计算机(其中码头工人安装)，请运行此命令：

```
#docker run 'cliqr/worker:latest' sleep 100000
```

```
Error: Unable to access jarfile cliqr-repository-client-*-jar-with-dependencies.jar^M
CLIQR_EXTERNAL_SERVICE_ERR_MSG_START^M
Failed to download files specified in environment variables^M
CLIQR_EXTERNAL_SERVICE_ERR_MSG_END^M
```

解决方案

1. CCO或码头工人计算机的洛金(如果独立)。
2. 保存旧有码头工人镜像。
3. 请使用root@abc-cco [] #码头工人保存cliqr/工作人员：新> oldimage.tar。
4. 删除码头工人镜像。

```
[root@abc-cco ~]# docker rmi -f a81630771eec
Untagged: cliqr/worker:latest
Deleted: sha256:a816567771eec9e8727162ab140b4e4a39fffwer43b3d403e56f2f12ed7c6d05f8
Deleted: sha256:167e65676beb12727aa74ddac2b7d87e113d1ee80cbecf93980595ca38cb92d37
Deleted: sha256:7f65225671815905d1b077b79c838f3fcff305a07dfg0129800605b104e7a71
Deleted: sha256:68f724567derd4031368a20e1ad72a15d0dfgdfg4ebf3dd5025932a2a625ee8a8c3
```

5. wget然后下载的新的码头工人文件。

```
root@abc-cco ~]#wget http://repo.cliqrtech.com/bin/docker/docker.tar
```

6. Untar码头工人镜像。

```
[root@abc-cco ~]# tar -xvf docker.tar
cliqr-container-worker/
cliqr-container-worker/Dockerfile
cliqr-container-worker/utils.sh
cliqr-container-worker/worker.sh
cliqr-container-worker/cliqr-repository-client-4.7.0-jar-with-dependencies.jar
```

```
[root@abc-cco ~]# cd cliqr-container-worker
```

7. 建立码头工人镜像。

```
[root@abc-cco cliqr-container-worker]# docker build -t 'cliqr/worker:latest' .
Sending build context to Docker daemon 73.26 MB
Step 1 : FROM centos:7
---> 0584b3d2cf6d
Step 2 : ENV JAVA_VERSION 1.7.0
---> Running in 5a30ce6e6e2f
---> ef6176cc1816
Removing intermediate container 5a30ce6e6e2f
Step 3 : RUN rpm -iUvh http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-9.noarch.rpm &&
sed -i "s/mirrorlist=https/mirrorlist=http/" /etc/yum.repos.d/epel.repo
&& yum clean all && yum -y update && yum -y install python-pip wget unzip ssh vim
&& yum -y install java-${JAVA_VERSION}-openjdk-headless
---> Running in f20c66af5d98

.....

.....

... output shorted for space
```

8. 一旦码头工人被重建，并且，当您运行此命令时，您能验证它：

```
[root@abc-cco cliqr-container-worker]# docker build -t 'cliqr/worker:latest' .
Sending build context to Docker daemon 73.26 MB
Step 1 : FROM centos:7
---> 0584b3d2cf6d
Step 2 : ENV JAVA_VERSION 1.7.0
---> Running in 5a30ce6e6e2f
---> ef6176cc1816
Removing intermediate container 5a30ce6e6e2f
Step 3 : RUN rpm -iUvh http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-9.noarch.rpm &&
sed -i "s/mirrorlist=https/mirrorlist=http/" /etc/yum.repos.d/epel.repo
&& yum clean all && yum -y update && yum -y install python-pip wget unzip ssh vim
&& yum -y install java-${JAVA_VERSION}-openjdk-headless
---> Running in f20c66af5d98

.....

.....

... output shorted for space
```

9. 实施与在CCO运行的外部生命周期操作的新应用。应用程序版本： ? 4.7.2