

# External Lifecycle Action Fails | Docker

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Problem](#)

[Verify](#)

[Solution](#)

## Introduction

This document describes how to resolve the issue when a user gets "Unable to access jar file cliqr-repository-client-\*-jar-with-dependencies.jar" Error.

## Prerequisites

## Requirements

There are no specific requirements for this document.

## Components Used

The information in this document is based on Cloud Center version 4.6 and above.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Problem

Usually, when you try to deploy an application with the use of External lifecycle action, Docker tries to execute the scripts initiated by the CCO. However, it fails because it cannot read the script itself due to the docker being corrupt.

### Error Message:

```
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)
```

**Note:** In GUI, it only shows with Error 4.

To rectify this issue, you need to follow these steps:

## Verify

From CCO/Docker machine (where docker is installed), run this command:

```
#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
```

## Solution

1. Login to CCO or docker machine (if standalone).
2. Save the old Docker image.
3. Use `root@abc-cco ~]# docker save cliqr/worker:latest > oldimage.tar`.
4. Remove the Docker image.

```
[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 the new docker file which is then downloaded.

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

6. Untar the Docker image.

```
[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. Build the Docker image.

```
[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. Once the docker is re-built, and when you run this command, you can verify it:

```
[root@abc-cco cliqr-container-worker]# docker run 'cliqr/worker:latest' sleep 100000
CLIQR_EXTERNAL_SERVICE_LOG_MSG_START
Download service bundle: success.
CLIQR_EXTERNAL_SERVICE_LOG_MSG_END
CLIQR_EXTERNAL_SERVICE_LOG_MSG_START
Executing service sleep action with command: "" from directory:
CLIQR_EXTERNAL_SERVICE_LOG_MSG_END
[root@abc-cco cliqr-container-worker]#
```

## 9. Deploy a new application with the External lifecycle action which runs on CCO. Application Version: 4.7.2