

FUJITSU Software

Interstage List Creator V10.6.0

Red Hat OpenShift上での 動作手順書

Linux(64)

CIB-2913-18-LWP0051(02)
2020年11月

はじめに

本資料は、Interstage List CreatorをRed Hat OpenShift V4上で動作させる手順について説明したものです。

なお、本資料の中では、ソフトウェアの名称を、以下のように省略して表記します。

ソフトウェア名称	略称
Interstage List Creator Enterprise Edition V10.6.0	List Creator
Interstage List Creator Connector V10.6.0	List Creator Connector
Red Hat OpenShift	OpenShift

前提知識

本資料を読む場合、以下の知識が必要です。

- Red Hat Enterprise Linuxに関する基本的な知識
- DockerおよびPodmanに関する基本的な知識
- Red Hat OpenShiftに関する基本的な知識
- Interstage List Creatorに関する基本的な知識
- UpdateAdvisor(ミドルウェア)に関する基本的な知識

検証環境

本資料の手順は、以下を使用して検証しています。

- Red Hat Enterprise Linux 8
- Red Hat OpenShift Container Platform 4
- Linux(Intel64)版 Interstage List Creator Enterprise Edition V10.6.0
- Linux(Intel64)版 Interstage List Creator Connector V10.6.0

商標

- Linux(R) は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- Red Hat(R)、Red Hat Enterprise Linux(R)、OpenShift(R)は米国およびその他の国において登録されたRed Hat, Inc.の商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- その他の記載されている商標および登録商標については、一般に各社の商標または登録商標です。

著作権

Copyright 2020 FUJITSU LIMITED

2020年11月 3版

目次

第1章 動作環境.....	1
1.1 対象製品.....	1
1.2 準備するもの.....	1
1.3 本資料の記載範囲.....	1
第2章 概要.....	2
第3章 コンテナイメージの作成.....	3
3.1 List Creatorのベアイメージの作成.....	3
3.1.1 List CreatorのパッケージDVDのマウント.....	3
3.1.2 インストールパラメーターCSVファイルの作成.....	3
3.1.3 Dockerfileの作成.....	4
3.1.4 コンテナイメージのビルド.....	5
3.1.5 List CreatorのパッケージDVDのアンマウント.....	5
3.2 List Creatorの環境設定とコンテナイメージの作成.....	5
3.2.1 ベアイメージからのコンテナの起動.....	5
3.2.2 List Creatorの緊急修正の適用.....	6
3.2.3 List Creatorの環境設定.....	6
3.2.4 コネクタ連携機能の環境構築(コネクタ連携機能を使用する場合).....	6
3.2.5 コンテナイメージの作成.....	7
第4章 Red Hat OpenShift上でのアプリケーションの実行.....	8
付録A トラブル情報の採取方法.....	10

第1章 動作環境

1.1 対象製品

本資料の対象製品は、以下です。

- Linux(Intel64)版 Interstage List Creator Enterprise Edition V10.6.0
- Linux(Intel64)版 Interstage List Creator Connector V10.6.0

1.2 準備するもの

List CreatorをOpenShift上で動作させるためには、事前に以下を準備してください。

- Red Hat Enterprise Linux 8
- Red Hat OpenShift Container Platform
- List CreatorのパッケージDVD(1枚目)
- UpdateAdvisor(ミドルウェア)およびList Creatorの最新の緊急修正

1.3 本資料の記載範囲

本資料は、OpenShift上でList Creatorを使用してPDFファイル保存を行うための手順を記載しています。ただし、以下は記載範囲外です。

- 他のコンテナとの連携方法
- 永続ストレージの利用方法

第2章 概要

OpenShift上でList Creatorを動作させるには、以下のことを行います。

1. List Creatorをインストールしたコンテナイメージを作成する

ポイント

.....
List Creatorをインストールした直後の状態のコンテナイメージです。本資料では、このコンテナイメージをList Creatorのベースイメージと呼びます。
.....

2. List Creatorの環境をセットアップしたコンテナイメージを作成する
3. OpenShiftでコンテナイメージを実行する

第3章 コンテナイメージの作成

3.1 List Creatorのベアイメージの作成

List Creatorのベアイメージは以下の手順で作成します。

- 以下の資材を作成し、同じディレクトリに配置します。以下の説明では、“\$HOME/container/lc”に配置しています。
 - List CreatorのパッケージDVDをマウントしたディレクトリ。以下の説明では“lc_mnt”としています。
 - List CreatorのインストールパラメーターCSVファイル。以下の説明では“lc_param.csv”としています。
 - Dockerファイル。以下の説明では、“Dockerfile”としています。
- コンテナイメージをビルドします。



List Creatorのベアイメージのサイズは約2.8GBです。

List Creatorのベアイメージをビルドする過程で、List CreatorのパッケージDVDの内容(約1GB)をコンテナイメージ内にコピーしますので、一時的に3.8GB程度のディスク容量が必要になります。本手順の実行は、ディスクの空き容量を十分確保した環境で行ってください。

3.1.1 List CreatorのパッケージDVDのマウント

List CreatorのパッケージDVDを、ローカルディスクに“lc_mnt”としてマウントします。マウント先は、この後説明するDockerfileの格納先と同じにします。パッケージDVDの内容はコンテナイメージ内にコピーされ、List Creatorのインストールに使用されます。

以下は、パッケージDVDを\$HOME/container/lc/lc_mntにマウントする例です。

- ローカルディスクにマウント先のディレクトリを作成します。

```
$ mkdir -p $HOME/container/lc/lc_mnt
```

- パッケージDVDをDVDドライブに挿入し、以下のコマンドで\$HOME/container/lc/lc_mntにマウントします。

```
$ sudo mount -t iso9660 -r /dev/cdrom $HOME/container/lc/lc_mnt
```



mountコマンドを実行するためには、システム管理者権限(root)が必要です。

3.1.2 インストールパラメーターCSVファイルの作成

図1の内容のインストールパラメーターCSVファイルを、ファイル名“lc_param.csv”で作成し、\$HOME/container/lcに格納します。インストールパラメーターCSVファイルの詳細については、List Creatorのインストールガイドを参照してください。

```
installInfo,softwareName,Interstage List Creator
installInfo,OS, Linux
installInfo,Version,V10.6.0
installInfo,Edition,Enterprise Edition
installInfo,Name,Interstage List Creator
parameters,szDir,/opt
```

3.1.3 Dockerfileの作成

図2はList Creatorのベースイメージを作成するためのDockerfileの例です。図2の下線部は環境に合わせて修正してください。Dockerfile内の各部分の意味については、後述の(1)～(11)の説明を参照してください。

```
# Get Base image ... (1)
FROM registry.access.redhat.com/ubi8:8.1
MAINTAINER <イメージの作者情報を示す任意の文字列>

# Set Japanese locale ... (2)
RUN yum clean all
RUN rm -f /etc/rpm/macros.image-language-conf && ¥
sed -i '/^override_install_langs=/d' /etc/yum.conf && ¥
yum -y reinstall glibc-common && ¥
yum -y install glibc-langpack-ja
env LANG=ja_JP.UTF-8 ¥
LC_ALL="ja_JP.UTF-8"
RUN yum -y reinstall tzdata && ¥
ln -snf /usr/share/zoneinfo/Asia/Tokyo /etc/localtime && ¥
sed -ri 's/en_US/ja_JP/' /etc/locale.conf

# Install rpms for List Creator ... (3)
RUN yum -y install glibc.x86_64 glibc.i686 nss-softokn-freebl.x86_64 nss-softokn-freebl.i686 ¥
libgcc.x86_64 libgcc.i686 libstdc++.x86_64 libstdc++.i686 redhat-lsb.x86_64 zip unzip ncompress ¥
libnsl.x86_64 libnsl.i686

# Install syslogd ... (4)
RUN yum install -y rsyslog
RUN sed 's/ModLoad imjournal/# ModLoad imjournal/' -i /etc/rsyslog.conf && ¥
sed 's/0mitLocalLogging on/0mitLocalLogging off/' -i /etc/rsyslog.conf && ¥
sed 's/IMJournalStateFile imjournal.state/# IMJournalStateFile imjournal.state/' -i /etc/rsyslog.conf

# Create directories
RUN mkdir /work

# Copy List Creator install DVD to the container image ... (5)
COPY ./lc_mnt /work/lc_mnt/

# Copy parameter csv file to the container image ... (6)
COPY ./lc_param.csv /work/lc_param.csv

# Copy silent install files to the container image ... (7)
RUN cp /work/lc_mnt/citool/install/RHEL6/* /work/

# Copy form data to the container image ... (8)
RUN mkdir /work/form && cp /work/lc_mnt/Samples/utf8/URIAGE.* /work/form/
RUN mkdir /work/form/kol5 && cp /work/lc_mnt/Samples/utf8/URIAGE.ovd /work/form/kol5/

# Install List Creator ... (9)
WORKDIR /work
RUN ./lc_silent_install.sh /work/lc_mnt lc_param.csv

# Specify executable when run this container image ... (10)
CMD [ "/usr/sbin/init" ]

# Remove List Creator installer ... (11)
WORKDIR /
RUN rm -fr /work/instparam /work/libcsv_parse.so /work/lc_silent_install.sh ¥
/work/lc_param.csv /work/lc_mnt
```

図2 List Creatorのベースイメージ用Dockerfileの例

- (1)ベースとなるコンテナイメージを指定します。
- (2)コンテナイメージのシステムロケール・タイムゾーンを日本に設定します。
- (3)List Creatorの必須パッケージ(rpm)をインストールします。必須パッケージについては、List Creatorのインストールガイド、「Red Hat(R) Enterprise Linux(R) 8に関する注意事項」を参照してください。「Red Hat(R) Enterprise Linux(R) 8に関する注意事項」は、以下のURLを参照してください。
<https://www.fujitsu.com/jp/products/software/resources/technical/interstage/listcreator/notes/>
- (4)コンテナ(pod)内にシステムログを出力する場合はsyslogdをインストールします。本手順を省略すると、システムログはコンテナ(pod)内に出力されません。なお、システムログが失われないように永続ストレージに出力することを推奨します。
- (5)List Creatorのインストール媒体を展開したディレクトリである“lc_mnt”をコンテナイメージにコピーします。COPYコマンドでは、./lc_mnt/*がコンテナイメージの/work/lc_mntにコピーされます。
- (6)3.1.2で作成したインストールパラメーターCSVファイルをコンテナイメージの/work/lc_param.csvにコピーします。
- (7)List Creatorのインストールスクリプトをコンテナイメージの/workにコピーします。
- (8)帳票出力のための資源をコンテナイメージの/work/formにコピーします。本資料では、サンプル帳票の“URIAGE”を使用しています。
- (9)サイレントモードでList Creatorをインストールします。
- (10)コンテナイメージを起動したときに実行されるコマンドを指定します。
- (11)不要になったファイルをコンテナイメージから削除します。

3.1.4 コンテナイメージのビルド

“podman build”コマンドでコンテナイメージをビルドします。

以下は、Dockerfileなどの資材格納ディレクトリが“\$HOME/container/lc”、ターゲット名が“lcee_bare”でビルドする例です。

```
$ podman build -t lcee_bare $HOME/container/lc
```

3.1.5 List CreatorのパッケージDVDのアンマウント

3.1.1でマウントしたList CreatorのパッケージDVDをアンマウントします。

```
$ sudo umount $HOME/container/lc/lc_mnt
```



umountコマンドを実行するためには、システム管理者権限 (root) が必要です。

3.2 List Creatorの環境設定とコンテナイメージの作成

3.1で作成したベアイメージを実行し、List Creatorの環境設定を行います。



コネクタ連携機能を使用する場合はアプリケーションサーバ(List Creator Connector)のベアイメージも必要です。3.1の“List Creator”を“List Creator Connector”に読み替えて、アプリケーションサーバのベアイメージも作成してから、本章の作業を行ってください。

3.2.1 ベアイメージからのコンテナの起動

必ず以下のオプションをつけて起動してください。

--privileged

以下は、コンテナイメージからコンテナを起動する例です。以下の例ではコネクタ連携で使用する帳票出力サーバのポートをホストの3913ポートに、アプリケーションサーバのポートをホストの3914ポートにマッピングしています。なお、スタンドアロン型での運用を行う場合は、ポート番号(-pオプション)を指定する必要はありません。

- 帳票出力サーバのコンテナの起動例(コンテナイメージ名:lcee_bare、コンテナ名:lcee_bare_c1、ホスト名:LCEE)

```
$ podman run --name lcee_bare_c1 --privileged -di --hostname=LCEE -p 3913:3913 lcee_bare
```

- アプリケーションサーバのコンテナの起動例(コンテナイメージ名:lcco_bare、コンテナ名:lcco_bare_c1、ホスト名:LCCO)

```
$ podman run --name lcco_bare_c1 --privileged -di --hostname=LCCO -p 3914:3914 lcco_bare
```

3.2.2 List Creatorの緊急修正の適用

“podman exec”コマンドでコンテナにログインし、UpdateAdvisor(ミドルウェア)のインストールおよび、List Creatorの最新の緊急修正の適用を行います。

- コンテナへのログイン例

```
$ podman exec -it lcee_bare_c1 /bin/bash
[root@LCEE /]#
```



参考

UpdateAdvisor(ミドルウェア)のインストール資材、および緊急修正モジュールは、“podman cp”コマンドを使用してコンテナに複写します。



参照

UpdateAdvisor(ミドルウェア)のインストール方法についてはUpdateAdvisor(ミドルウェア)のヘルプを参照してください。

List Creatorの緊急修正の適用方法については、緊急修正の修正情報ファイル、およびUpdateAdvisor(ミドルウェア)のヘルプを参照してください。

3.2.3 List Creatorの環境設定

List Creatorの環境設定を行います。



参考

帳票格納ディレクトリ、監査証跡ログの格納ディレクトリなどは、情報が失われないように永続ストレージを使用することを推奨します。



参照

List Creatorの環境設定の詳細については、List Creatorのオンラインマニュアル「環境設定・帳票運用編」を参照してください。

3.2.4 コネクタ連携機能の環境構築(コネクタ連携機能を使用する場合)

List Creatorのコネクタ連携機能を使用する場合は、コネクタ連携機能のセットアップ用シェルを実行します。

- 帳票出力サーバの例

```
[root@LCEE /]# cd /opt/FJSVoast/remoteprint
[root@LCEE /]# sh setupFOS.sh
```

- アプリケーションサーバの例

```
[root@LCCO /]# cd /opt/FJSVoast/remotepint  
[root@LCCO /]# sh setupAPS.sh
```



.....
セットアップ用シェルの詳細については、オンラインマニュアルの「環境設定・帳票運用編」を参照してください。
.....

3.2.5 コンテナイメージの作成

以下の手順で、コンテナイメージをコミットし、新しいコンテナイメージを作成します。

1. 次回のコンテナの起動に不要な以下のファイルを削除し、コンテナからexitします。

- シスログの出力先 (/var/log/messages)

```
[root@LCEE /]# rm -f /var/log/messages  
[root@LCEE /]# exit
```

2. List Creatorの環境設定を行ったコンテナをコミットし、新しいコンテナイメージを作成します。

- 新しいコンテナイメージ名がlceeappの場合の例

```
$ podman commit lcee_bare_c1 lceeapp
```

3. コンテナを停止し、削除します。

- コンテナ名がlcee_bare_c1、コンテナIDがabcdef123456の場合の例

```
$ podman stop lcee_bare_c1  
abcdef123456  
$ podman rm lcee_bare_c1  
abcdef123456
```

第4章 Red Hat OpenShift上でのアプリケーションの実行

以下の手順を実施して、OpenShift上でList Creatorを使用して帳票をPDFファイル保存します。

参考

- List Creatorを動作させるコンテナは、root権限で特権コンテナとして実行する必要があります。root権限、かつ特権コンテナとしてコンテナイメージを実行できるように、OpenShiftのユーザ、およびプロジェクトの設定をしてください。
- 本章では帳票出力サーバでの操作手順を説明していますが、コネクタ連携機能を使用する場合は、アプリケーションサーバにおいても同様の操作を行います。
- 帳票資源の格納先や帳票の保存先は、情報が失われないように永続ストレージを使用することを推奨します。

1. OpenShiftの内部レジストリにログインします。以下は、ユーザ名がreguser、パスワードがregpassword、内部レジストリのIPアドレスが192.168.100.102、ポート番号が5000の例です。tls非対応の内部レジストリを使う場合は、podman loginに--tls-verify=falseを追加します。

```
$ podman login -u reguser -p regpassword --tls-verify=false 192.168.100.102:5000
```

2. OpenShiftの内部レジストリに、3.2で作成したコンテナイメージを登録します。

本手順では、コンテナイメージを作成した環境で実行します。

以下は、OpenShiftの内部レジストリのIPアドレスが192.168.100.102、ポート番号が5000の例です。イメージ名“lceeapp”、タグ名“latest”で登録しています。tls非対応の内部レジストリを使う場合は、podman pushに--tls-verify=falseを追加します。

```
$ podman tag lceeapp 192.168.100.102:5000/lceeapp:latest
$ podman push --tls-verify=false 192.168.100.102:5000/lceeapp:latest
```

3. OpenShiftにログインします。以下は、ユーザ名がosuser、パスワードがospasswordの例です。

```
$ oc login -u osuser -p ospassword
```

4. OpenShiftの内部レジストリに登録したコンテナイメージを取得して、pod上で動くList Creatorのテンプレートを作成します。以下の例では、lceeapp.yamlというファイル名のテンプレート(yaml形式)が出力されます。

```
$ oc new-app -i lceeapp:latest -o yaml > lceeapp.yaml
```

5. 4で出力されたテンプレートを編集して、以下の設定を行います。

- List Creatorが動作するコンテナを特権コンテナとして起動する。
- Serviceの設定を行う。(コネクタ連携機能を使用する場合)

“securityContext”(下線部)を、DeploymentConfigのspec.template.specの中に追加します。下記の例では、SCC(SEcurity CONTEXT CONSTRAINTS)が割り当てられたサービスアカウント“privsvacct”を指定しています。インデントは以下の例に合わせてください。

```
~~~抜粋開始~~~
spec:
  containers:
  - image: ''
    name: lceeapp
    ports:
    - containerPort: 3913
      protocol: TCP
    resources: {}
    securityContext:
      privileged: true
    securityContext:
```

```
runAsUser: 0
serviceAccount: privsvcaact
serviceAccountName: privsvcaact
~~抜粋終了~~
```

コネクタ連携機能を使用する場合は、必要に応じてServiceの設定を行います。下記の例では、ServiceのtypeをNodePortにし、30001ポートをnodePortとして設定しています。

```
~~抜粋開始~~
spec:
  type: NodePort
  ports:
  - name: 3913-tcp
    port: 3913
    protocol: TCP
    targetPort: 3913
    nodePort: 30001
~~抜粋終了~~
```

- 5で編集したテンプレートを 사용하여 コンテナを起動します。

```
$ oc create -f lceeapp.yaml
```

- コンテナ (Pod) の情報を確認します。以下は、コンテナ (Pod) 名が "lceeapp-xxxxxxxxxxxxxxxx" の例です。

```
$ oc get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/lceeapp-xxxxxxxxxxxxxxxx        1/1     Running   0           8s

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/lceeapp                     NodePort      zzz.zzz.zzz.zzz <none>        3913:30001/TCP   9s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/lceeapp              1/1     1             1           9s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/lceeapp-yyyyyyyyyy  1         1         1       8s

NAME                                IMAGE REPOSITORY          TAGS    UPDATED
imagestream.image.openshift.io/lceeapp  192.168.100.102:5000/lceeapp  latest  19 seconds ago
```

- 7で確認したコンテナ (Pod) にログインします。

```
$ oc rsh lceeapp-xxxxxxxxxxxxxxxx
sh-4.4#
```

- 帳票出力コマンド (prprint) を使用して、帳票をPDFファイル保存します。

```
sh-4.4# PATH=/opt/FJSVoast/bin:$PATH
sh-4.4# export PATH
sh-4.4# LD_LIBRARY_PATH=/opt/FJSVoast/lib:$LD_LIBRARY_PATH
sh-4.4# export LD_LIBRARY_PATH
sh-4.4# prprint URIAGE -assetdir /work/form/ -f /opt/FJSVoast/data/utf8/URIAGE.dat -atdirect file -keeppdf /var/tmp/URIAGE.pdf
sh-4.4#
```

- 保存されたPDFファイル (/var/tmp/URIAGE.pdf) を確認します。

付録A トラブル情報の採取方法

ここでは、OpenShift上でトラブル調査用の情報を採取する方法について説明します。

OpenShift上で起動したコンテナ(Pod)上では、障害調査用情報採取ツール(OARSdtコマンド)を使用してトラブル情報を採取します。

注意

Red Hat OpenShift Container Platform 4で起動したコンテナ(pod)上では、FJQSS(資料採取ツール)は利用できません。

FJQSSと同じ情報を採取できる障害調査用情報採取ツール(OARSdtコマンド)を使用してください。

参照

障害調査用情報採取ツール(OARSdtコマンド)の詳細については、オンラインマニュアルの「トラブルシューティング集」を参照してください。

以下に、トラブル情報の採取例を記載します。

1. コンテナ(Pod)名を確認します。以下は、コンテナ(Pod)名が"lceeapp-xxxxxxxxxxxxxxxx"の例です。

```
$ oc get pod
NAME                                READY  STATUS   RESTARTS  AGE
pod/lceeapp-xxxxxxxxxxxxxxxx        1/1    Running  0          8s
```

2. 1で確認したコンテナ(Pod)にログインします。

```
$ oc rsh lceeapp-xxxxxxxxxxxxxxxx
sh-4.4#
```

3. 障害調査用情報採取ツール(OARSdtコマンド)を使用して、トラブル調査用の情報を採取します。

```
sh-4.4# /opt/FJSVoast/bin/OARSdt /var/tmp
The data is collected.
.....
The archive file is created.
The archive file is compressed.
OARSdt was completed.
sh-4.4#
```

4. コンテナ(pod)からログアウトし、トラブル調査用の情報を取得します。

```
sh-4.4# exit
exit
$ oc cp lceeapp-xxxxxxxxxxxxxxxx:var/tmp/oapress.cpio.gz ./oapress.cpio.gz
```