

WebSAM FileTransfer コンテナ利用手順書

改版履歴

版数	改版履歴	内容
1	2019/09/30	新規作成

免責事項

本書の内容は、予告なしに変更されることがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任をおいません。

また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、お客様の責任とさせていただきます。

本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

商標情報

WebSAM は日本電気株式会社の登録商標です。

本書に記載された製品名および標語は、各社の商標または登録商標です。

本書に記載されたシステム名、社名、製品名等はそれぞれの会社の商標および登録商標です。

目次

WebSAM FileTransfer	1
1 章. はじめに.....	5
2 章. コンテナイメージ作成.....	6
2.1. Docker による作成	6
2.1.1. DockerFile を使用してイメージを作成する手順.....	6
2.1.2. Docker コンテナを起動してから構築、イメージを作成する手順.....	8
2.2. OpenShift による作成.....	11
3 章. コンテナイメージ利用方法.....	15
3.1. Docker での利用	15
3.2. OpenShift での利用.....	16
4 章. 注意・補足事項.....	21

1章. はじめに

本書は、WebSAM FileTransfer（以下、FT）をコンテナ環境で利用するための手順書です。

この手順は、Docker および OpenShift を使用する手順となります。

この手順書では、OpenShift によって作成したコンテナイメージを Docker で利用するようには対応していません。Docker での利用を予定している場合は、手順「2.1. Docker による作成」を参照してください。

OpenShift での利用を予定している場合は、手順「2.1. Docker による作成」または、手順「2.2. OpenShift による作成」のどちらで作成しても対応しております。

本書で使用する用語について以下に説明します。

用語	説明
DockerFile	DockerFile とは、通常、イメージを構築するために手動で実行が必要なコマンドが、テキスト形式で列挙されたファイルです。これを読み込ませることで、自動でイメージを構築できます。
レジストリ	レジストリ（registry）とはイメージを持つリポジトリを格納、管理するサービスです。
リポジトリ	リポジトリ（repository）とは Docker イメージの集まりです。リポジトリの中では、イメージの違いをタグでラベル付けします。
タグ	タグ（tag）は リポジトリ 上の Docker イメージに割り当てるラベルです。タグを使い、リポジトリ上のイメージを識別できるようにします。
イメージ	イメージ（image）とはルート・ファイルシステムに対する変更を並べ集めたもので、コンテナを実行する際に使われる実行パラメータに相当します。
プロジェクト	OpenShift は、同一の物理クラスター上で複数の仮想クラスターの動作をサポートしています。この仮想クラスターをプロジェクトと呼びます。
デプロイ	OpenShift で、コンテナイメージをポッドとして利用可能な状態にすることです。
ポッド	1 つ以上のコンテナから成る OpenShift 上で扱う最小の単位となります。 同一のポッドに含まれるコンテナはネットワークインタフェース等を共有します。

[対象製品]

- WebSAM FileTransfer Ver5.3

[対象 OS]

- RedHat Enterprise Linux 7

2章. コンテナイメージ作成

この章では、コンテナイメージの作成方法を説明します。

Docker コマンドを使用した方法と、OpenShift のコマンドラインインターフェース、oc コマンドを使用した方法について説明します。

2.1. Docker による作成

Docker による作成方法については、DockerFile を使用方法と、ベースイメージを使用して構築していく方法の 2 通り説明します。

2.1.1. DockerFile を使用してイメージを作成する手順

(1) DockerFile 等の必要なファイルを用意

DockerFile を格納するディレクトリを作成し、DockerFile、および FT のインストール媒体、実行するスクリプト等を格納します。

ここでは次のコマンドで /home/FileTransfer に docker というディレクトリを作成し、必要なファイルを格納します。

```
# cd /home/FileTransfer
# mkdir docker
```

■必要なファイル

ファイル名	詳細
Dockerfile *1	コンテナを構築するために必要な以下のファイルを配置、実行するように設定された DockerFile です。
FT53RED	FT のインストール媒体。
esmpo_start.sh *2	OpenShift の環境で FT を実行可能にするため、レジストリエディタを使用して、root ユーザ以外の権限で FT を起動するための設定を実施します。
ftsetup.sh *3	FT をインストールするためのスクリプト。FT の依存パッケージのインストールおよび FT のインストールを実施します。
ft_daemon	FT の起動/停止スクリプト。一般ユーザで起動できるように変更。
t_ft_daemon	FT の起動/停止スクリプト。ft_serv をフォアグラウンドで実行するように変更。
yumadd.txt	yum 実行時のプロキシサーバ情報ファイル。 *4

*1 Dockerfile の 1 行目、"FROM baseimage:latest"となっている所は、実際に利用するベースイメージに合わせて編集してください。ファイルの詳細は「4 章.注意補足事項」の「・ DockerFile の実行内容」を参照してください。

*2 ファイルの詳細は「4 章.注意補足事項」の「・ esmpo_start.sh の内容」を参照してください。

*3 ファイルの詳細は「4 章.注意補足事項」の「・ ftsetup.sh の内容」を参照してください。

*4 プロキシサーバ情報ファイル yumadd.txt はデフォルトでは以下の内容となっているため、実際のプロキシサーバの環境に合わせて編集してください。

■yumadd.txt 内容

```
proxy=http://proxy.example.nec.co.jp:8080
```

(2) コンテナのイメージをビルド

次のコマンドを実行し、イメージをビルドします。

```
# docker build -t ftimage01:ft_installed /home/FileTransfer/docker/
```

“-t”オプションでイメージにリポジトリ名とタグを付けます。続いて DockerFile 等を格納したディレクトリのパスを指定します。

例では「ftimage01」というリポジトリ名で「ft_installed」というタグを付けたイメージをビルドしています。“ft_installed”の部分は省略可能で、省略すると自動で「latest」というタグが付きます。

(3) ビルドの結果を確認

次のコマンドを実行し、「REPOSITORY」「TAG」の列を参照し、ビルドしたイメージが表示されることを確認します。

```
# docker images
```

■ コマンドの実行結果

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ftimage01	ft_installed	67bdba0cf167	About an hour ago	440MB

「REPOSITORY」列にはリポジトリ名が、「TAG」列にはタグが、「IMAGE ID」列にはイメージ ID が、「CREATED」列には作成されてからの経過時間が、「SIZE」列にはコンテナのサイズが表示されます。

※「IMAGE ID」はランダムで払い出されます。

2.1.2. Docker コンテナを起動してから構築、イメージを作成する手順

(1) FT のインストール媒体を用意

任意のディレクトリに FT のインストール媒体を用意します。

ここでは「/home/FileTransfer」にインストール媒体「FT53RED」を格納します。

(2) コンテナの起動

ベースとなるイメージを用意し、そのイメージからコンテナを起動します。

次のコマンドを実行し、コンテナを起動します。

```
# docker run -d -v /home/FileTransfer:/share/FileTransfer --name ftcon01 baseimage
```

“-d”オプションでコンテナをバックグラウンドで起動します。

“-v” オプションでホストマシンのディレクトリを、コンテナのディレクトリにマウントします。

ここではホスト「/home/FileTransfer」をコンテナ「/share/FileTransfer」にマウントします。

“--name”オプションで起動するコンテナの名前を指定します。ここでは「ftcon01」という名前を指定します。

続いてベースとなるイメージのイメージ名を指定します。

(3) コンテナへのアクセス

次のコマンドを実行しコンテナ「ftcon01」へアクセスします。

```
# docker exec -it ftcon01 /bin/bash
```

“-i”オプションで標準入力を開き続け、“-t”オプションで疑似 tty を割り当てます。

“-i”オプションと“-t”オプション両方を指定した上で、“/bin/bash”を実行することでコンテナを操作できるようになります。

(4) プロキシサーバの情報を登録

ここからコンテナを操作します。

コンテナへのアクセスに成功した場合、マシン名の表示がコンテナ ID へと変わります。コンテナ ID はランダムに払い出されますので、” 67bdba0cf167 ” の部分はコンテナにより異なります。

次のコマンドでファイル「/etc/yum.conf」を編集します。

```
[root@67bdba0cf167 /]# vi /etc/yum.conf
```

ファイルに次のようなプロキシサーバのエントリを追加し、yum コマンドを利用可能にします。

■ 追加するエントリ（実際のプロキシサーバの環境に合わせて追加してください。）

```
proxy=http://proxy.example.nec.co.jp:8080
```

(5) モジュールを最新化

次のコマンドを実行し、モジュールを最新の状態に更新します。

```
[root@67bdba0cf167 /]# yum -y update
```

(6) FT のインストールに必要なパッケージをインストール

次のコマンドを実行し、FT のインストールに必要なパッケージをインストールします。

```
[root@67bdba0cf167 /]# /bin/yum -y install glibc.i686 libgcc.i686 libstdc++.i686
```

(7) FT をインストール

次のコマンドを実行し、FT のインストールを行います。

```
[root@67bdba0cf167 /]# rpm -ivh /share/FileTransfer/FT53RED
```

(8) exit でコンテナの操作を終了

FT のインストールや、その他必要な設定が完了したら次のコマンドでコンテナの操作を終了します。

```
[root@67bdba0cf167 /]# exit
```

(9) コンテナの停止

ここからホストマシンの操作に戻ります。次のコマンドでコンテナ「ftcon01」を停止します。

```
# docker stop ftcon01
```

(10) コンテナの変更を保存

次のコマンドでこれまでに実施したコンテナ「ftcon01」に対する変更を保存します。

```
# docker commit ftcon01
```

(11) イメージ ID を取得

「docker commit」を実行すると、リポジトリ名とタグの付いていない新しいイメージが作成されます。次のコマンドで、作成された新しいイメージのイメージ ID を取得します。

```
# docker images
```

■ コマンドの実行結果

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	321e4bc6a955	About a minute ago	434MB
baseimage	latest	9f38484d220f	6 months ago	202MB

「REPOSITORY」列にはリポジトリ名が、「TAG」列にはタグが、「IMAGE ID」列にはイメージ ID が、「CREATED」列には作成されてからの経過時間が、「SIZE」列にはコンテナのサイズが表示されます。新しいイメージにはリポジトリ名とタグ付いていないので、該当列が「<none>」と表示されます。「<none>」と表示されている行の「IMAGE ID」を参照します。

※「IMAGE ID」はランダムで払い出されます。

(12) リポジトリ、タグの設定

次のコマンドで、リポジトリ名とタグの付いていないイメージにタグを付けます。

ここでは、手順(11)で取得したイメージ ID「321e4bc6a955」に対して、「ftimage01」というタグを付けます。

```
# docker tag 321e4bc6a955 ftimage01
```

(13) イメージの確認

次のコマンドで、リポジトリ名とタグが付いていることを確認します。

```
# docker images
```

■ コマンドの実行結果

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ftimage01	latest	321e4bc6a955	About a minute ago	434MB
baseimage	latest	9f38484d220f	6 months ago	202MB

2.2. OpenShift による作成

(1) DockerFile 等の必要なファイルを用意

DockerFile を格納するディレクトリを作成し、DockerFile、および FT のインストール媒体、実行するスクリプト等を格納します。ここでは次のコマンドで「/home/FileTransfer」に「docker」というディレクトリを作成し、必要なファイルを格納します。

```
# cd /home/ FileTransfer  
# mkdir docker
```

■必要なファイル

ファイル名	詳細
Dockerfile *1	コンテナを構築するために必要な以下のファイルを配置、実行するように設定された DockerFile です。
FT53RED	FT のインストール媒体。
esmpo_start.sh *2	OpenShift の環境で FT を実行可能にするため、レジストリエディタを使用して、root ユーザ以外の権限で FT を起動するための設定を実施します。
ftsetup.sh *3	FT をインストールするためのスクリプト。FT の依存パッケージのインストールおよび FT のインストールを実施します。
ft_daemon	FT の起動/停止スクリプト。一般ユーザで起動できるように変更。
t_ft_daemon	FT の起動/停止スクリプト。ft_serv をフォアグラウンドで実行するように変更。
yumadd.txt	yum 実行時のプロキシサーバ情報ファイル。 *4

*1 Dockerfile の 1 行目、「FROM baseimage:latest」となっている所は、実際に利用するベースイメージに合わせて編集してください。ファイルの詳細は「4 章.注意補足事項」の「・ DockerFile の実行内容」を参照してください。

*2 ファイルの詳細は「4 章.注意補足事項」の「・ esmpo_start.sh の内容」を参照してください。

*3 ファイルの詳細は「4 章.注意補足事項」の「・ ftsetup.sh の内容」を参照してください。

*4 プロキシサーバ情報ファイル yumadd.txt はデフォルトでは以下の内容となっているため、実際のプロキシサーバの環境に合わせて編集してください。

■yumadd.txt 内容

```
proxy=http://proxy.example.nec.co.jp:8080
```

(2) Openshift にログイン

次のコマンドで OpenShift にログインします。

```
# oc login -u ft-user openshift.example.nec.co.jp:443
```

“-u”オプションでログインユーザを指定します。続いてサーバを指定します。

ユーザ名、サーバの指定は実際の環境に合わせて指定してください。

(3) プロジェクトの作成

コマンドか、Web コンソールから「project」の作成ができます。

a. コマンドを使用する場合

次のコマンドを実行し、新しいプロジェクトを作成します。

ここでは「ft-project」を作成します。

```
# oc new-project ft-project
```

作成したプロジェクトは次のコマンドで確認できます。

```
# oc get projects
```

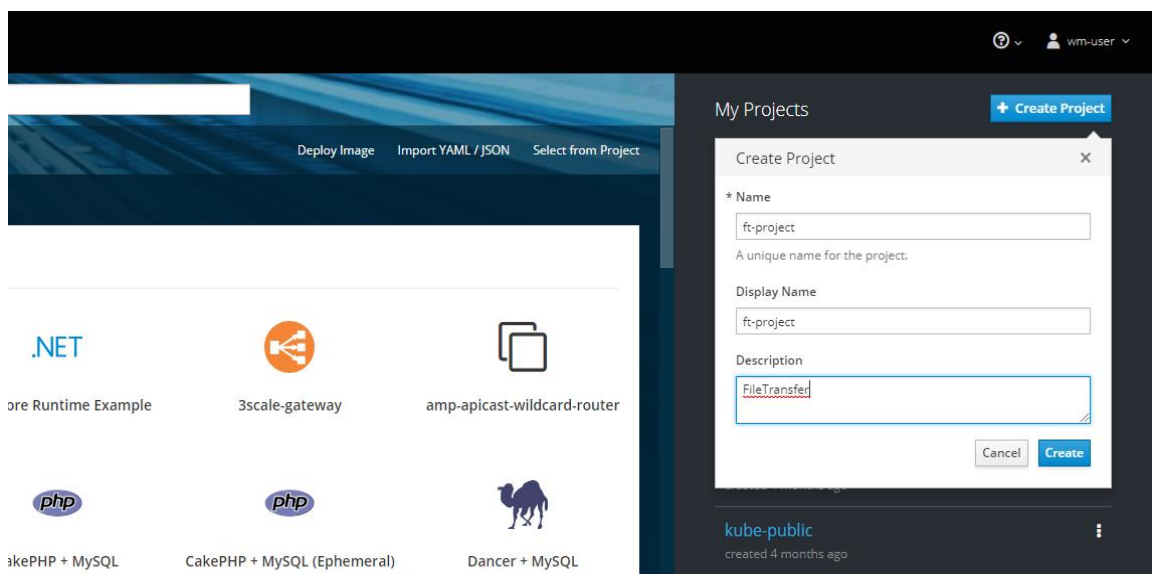
■ コマンドの実行結果

NAME	DISPLAY NAME	STATUS
ft-project	ft-proftcts	Active

「NAME」列にはプロジェクト名が、「DISPLAY NAME」列にはプロジェクトの表示名が、「STATUS」列には状態が表示されます。

b. Web コンソールを使用する場合

Web コンソールにアクセスし、画面右上の「Create Project」を選択し、「Name」「Display Name」「Description」を入力し、「Create」を選択します。



(4) プロジェクトの指定

次のコマンドで作業するプロジェクトを指定します。

```
# oc project ft-project
```

(5) 「build」の作成

次のコマンドで「build」を作成します。

```
# oc new-build --name=ftimage01 --strategy=docker --binary
```

“--name”オプションで「build」の名前を指定します。これがイメージの名前となります。

“--strategy”オプションでビルドの方法を指定します。ここでは DockerFile を使用しますので、「docker」を指定します。

“--binary”オプションでローカルのファイルを使用することを指定しています。

(6) 「build」の確認

コマンドか、Web コンソールから「build」の確認ができます。

a. コマンドを使用する場合

次のコマンドを実行し、「NAME」列に作成した「build」が存在することを確認します。

```
# oc get build
```

■ コマンドの実行結果

NAME	TYPE	FROM	STATUS	STARTED	DURATION
ftimage01-1	Docker	Binary	Complete	About a minute ago	6m44s

「NAME」列には“build”の名前が、「TYPE」列にはビルドの方法が、「FROM」列には参照する場所が、「STATUS」列には build の状態が、「STARTED」列には作成されてからの時間が、「DURATION」列には、作成にかかった時間がそれぞれ表示されます。

b. Web コンソールを使用する場合

Web コンソールにアクセスし、任意のプロジェクトを開きます。左メニューの「Builds」から「Builds」を選択し、作成した「build」が存在することを確認します。

The screenshot shows the OpenShift Container Platform web console. The top navigation bar includes the 'OPENSIFT CONTAINER PLATFORM' logo, 'Application Console', and user information. The left sidebar shows the 'ft-project' namespace selected, with a menu containing 'Overview', 'Applications', 'Builds', 'Resources', and 'Storage'. The 'Builds' menu item is highlighted. The main content area displays the 'Builds' page for the 'ft-project' namespace. It includes a search bar and a table with the following data:

Name	Last Build	Status	Duration	Created	Type	Source
ftimage01	No builds				Docker	Binary

(7) ビルドの実行

次のコマンドでビルドを実行します。

```
# oc start-build ftimage01 --from-dir=/share/FileTransfer/ --follow
```

“--from-dir”オプションで DockerFile 等を格納したディレクトリのパスを指定します。

“--follow”オプションはビルドの進捗や結果を画面に出力します。

(8) ビルド結果の確認

コマンドか、Web コンソールからビルド結果の確認ができます。

a. コマンドから確認する場合

次のコマンドを実行し、「NAME」列に作成したイメージが存在することを確認します。

```
# oc get is
```

■ コマンドの実行結果

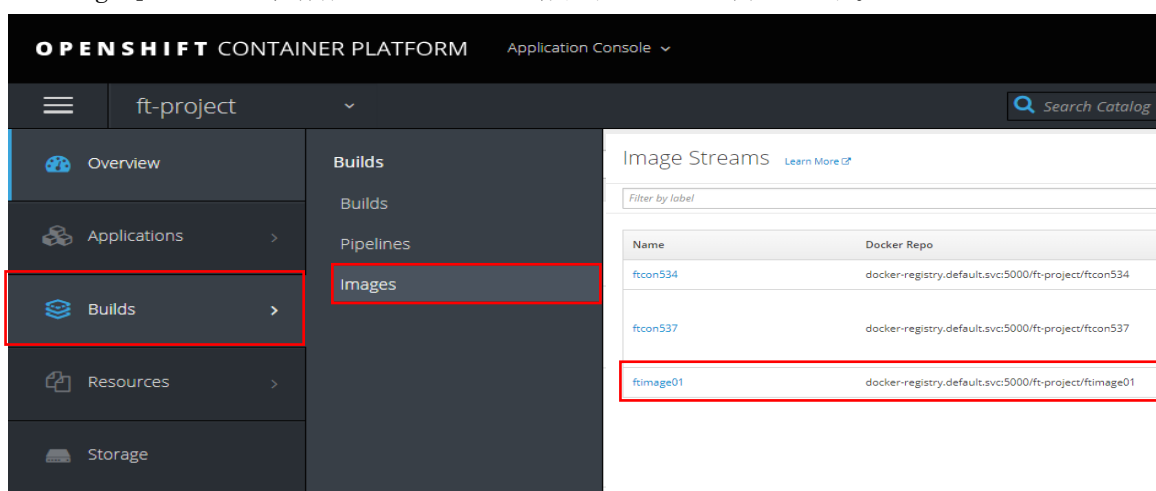
NAME	DOCKER REPO	TAGS	UPDATED
ftimage01	docker-registry.default.svc:5000/ft-project/ftimage01	latest	About a minute ago

「NAME」列にはイメージの名前が、「DOCKER REPO」列にはリポジトリ名が、「TAG」列にはタグが、「UPDATED」列にはビルドからの経過時間が表示されます。

※”oc start-build”で作成されたイメージは、自動で OpenShift の内部レジストリに登録されます。したがってリポジトリ名とタグについても自動で設定されることになります。リポジトリ名に設定されるレジストリ名、プロジェクト名は環境により異なります。実際の環境に読み替えてください。

b. Web コンソールを使用する場合

Web コンソールにアクセスし、任意のプロジェクトを開きます。左メニューの「Builds」から「images」を選択し、作成したイメージが存在することを確認します。



3章. コンテナイメージ利用方法

この章では、2章で作成したコンテナイメージを起動させる方法の説明をします。

Docker コマンドを使用した方法と、OpenShift のコマンドラインインターフェース、Web コンソールを使用した方法について説明します。

3.1. Docker での利用

手順 2.2. OpenShift による作成で作成したイメージについては Docker での利用はできません。

この手順では、手順 2.1. Docker による作成で作成したイメージを対象としています。

(1) コンテナの起動

次のコマンドを実行し、手順 2.1. で作成したイメージでコンテナを起動します。

```
# docker run -d -p 12000:2000 -p 13935:3935 --name ftcon01 ftimage01
```

“-d”オプションでコンテナをバックグラウンドで起動します。

“-p”オプションでホストマシンのポートとコンテナのポートをバインドします。

“-p”オプションは「-p <ホストマシンのポート>:<コンテナのポート>」のように指定します。

“-name”オプションでコンテナの名前を指定します。省略した場合ランダムな名前に設定されます。

続いて、起動するコンテナイメージを指定します。

ここでは「ftimage01」というコンテナイメージで「ftcon01」という名前のコンテナを起動します。

(2) コンテナの起動を確認

次のコマンドで、コンテナが起動していることを確認します。

```
# docker ps
```

■ コマンド実行結果

CONTAINER ID	IMAGE	COMMAND	CREATED
9ebe735c45c0	ftimage01	"/bin/sh -c '/opt/SS..."	10 seconds ago
	STATUS	PORTS	NAMES
	Up 7 seconds	0.0.0.0:12000->2000/tcp, 0.0.0.0:13935->3935/tcp	ftcon01

「CONTAINER ID」列にはコンテナ ID が、「IMAGE」列には使用したイメージ名が、「COMMAND」列にはコンテナの起動コマンドが、「CREATED」列には作成してから経過時間が、「STATUS」列にはコンテナの状態が、「PORTS」列にはバインドしているポートが、「NAMES」列にはコンテナ名が表示されます。

※「CONTAINER ID」はランダムに払い出されます。

3.2. OpenShift での利用

OpenShift において、コンテナは 1 つまたは複数のアプリケーションのコンテナから成るポッドという単位で起動されます。

OpenShift に手順「2.1. Docker による作成」で作成したような、既存の Docker イメージをデプロイする場合、docker コマンドを利用し、OpenShift の内部レジストリに向けてイメージをプッシュする必要があります。

手順「2.2. OpenShift による作成」で作成したイメージをデプロイする場合、イメージは既に OpenShift の内部レジストリに登録されているため、手順(4)から進めてください。

(1) タグを付ける

次のコマンドを実行し、既存のイメージにリポジトリ名とタグを付けます。

ここで付けたリポジトリ名とタグにより、どのレジストリにプッシュされるかが決定されます。

```
# docker tag ftimage01 docker-registry.default.svc:5000/ft-project/ftimage01
```

※レジストリ名、プロジェクト名は環境により異なります。実際の環境に合わせて設定してください。

例のコマンドの詳細は次のようになっています。

```
# docker tag <既存 image 名> <registry 名:ポート番号>/<project 名>/<image 名>
```

(2) イメージをプッシュ

次のコマンドを実行し、手順(1)でタグ付けしたイメージをプッシュします。

```
# docker push docker-registry.default.svc:5000/ft-project/ftimage01
```

(3) イメージの確認

プッシュしたイメージを確認します。

コマンドか、Web コンソールからイメージの確認ができます

a. コマンドから確認する場合

次のコマンドを実行し、「NAME」列に作成したイメージが存在することを確認します。

```
# oc get is
```

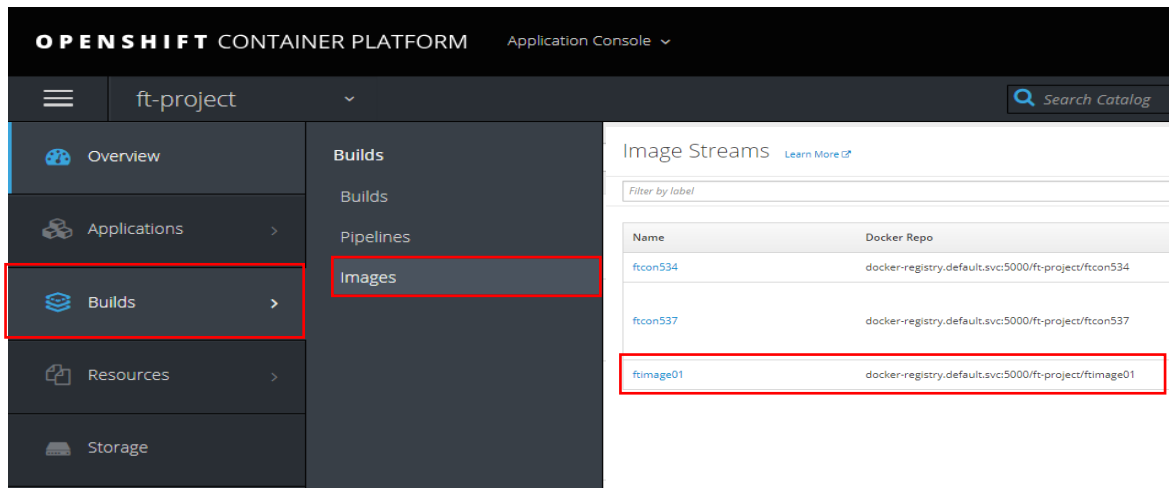
■ コマンドの実行結果

NAME	DOCKER REPO	TAGS	UPDATED
ftimage01	docker-registry.default.svc:5000/ft-project/ftimage01	latest	3 days ago

「NAME」列にはイメージの名前が、「DOCKER REPO」列にはリポジトリ名が、「TAG」列にはタグが、「UPDATED」列にはビルドからの経過時間が表示されます。

b. Web コンソールを使用する場合

Web コンソールにアクセスし、任意のプロジェクトを開きます。左メニューの「Builds」から「images」を選択し、作成したイメージが存在することを確認します。



(4) イメージのデプロイ

OpenShift の内部レジストリにプッシュしたイメージをデプロイします。

コマンドか、Web コンソールから実行できます

a. コマンドから実行する場合

次のコマンドを実行し、イメージのデプロイを実行します。

```
# oc new-app ftimage01 --name ftcon01
```

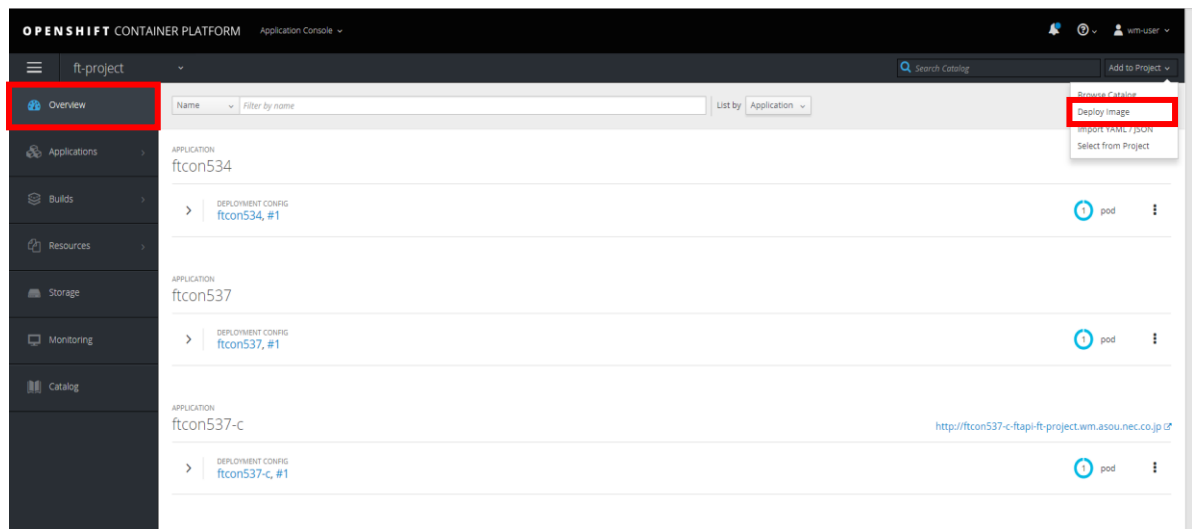
「oc new-app」に続いてイメージ名を指定します。

“--name”オプションで起動するポッドの名前を指定します。

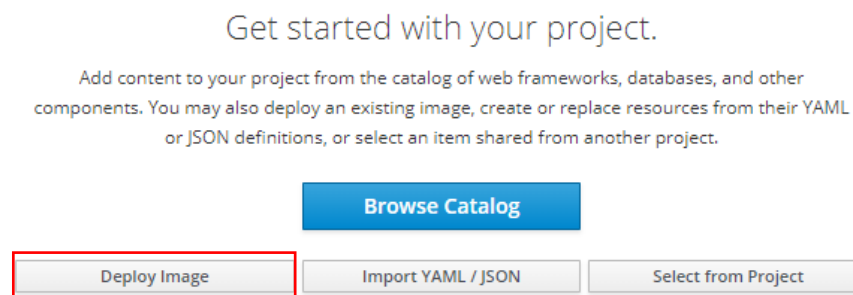
b. Web コンソールを使用する場合

b-1. Web コンソールにアクセスし、任意のプロジェクトを開きます。左メニューの「Overview」を選択します。

既に 1 つ以上のイメージがデプロイされている場合は、画面右上のプルダウンから「Deploy Image」を選択します。



b-2. 「Deploy Image」を選択します。



- b-3. ラジオボタン「Image Stream Tag」を選択したら、プルダウンからプッシュしたイメージを選択し、「Deploy」を選択します。
- b-4. 「*Name」から、ポッドの名前を指定できます。

(5) コンテナの確認

デプロイしたコンテナを確認します。

コマンドか、Web コンソールからイメージの確認ができます

a. コマンドから確認する場合

次のコマンドを実行し、「NAME」列にデプロイしたポッド名が存在することを確認します。

```
# oc get pod
```

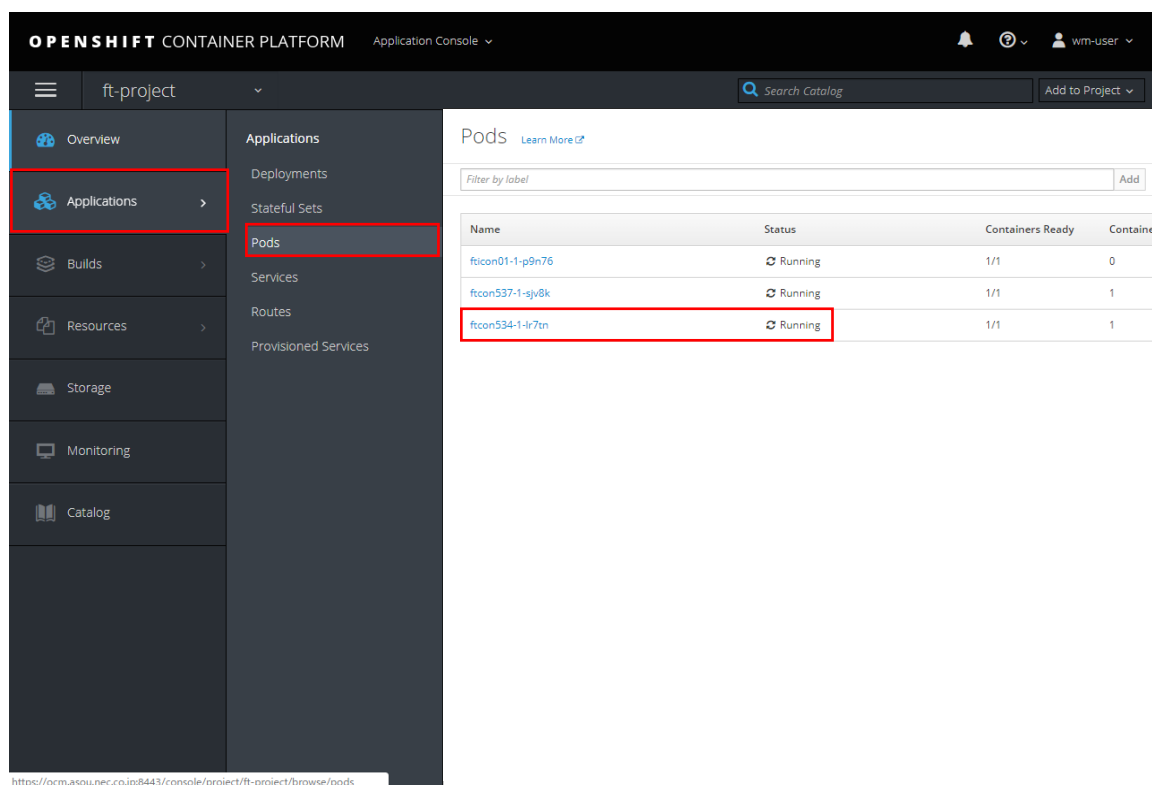
■ コマンドの実行結果

NAME	READY	STATUS	RESTARTS	AGE
ftcon01-1-jhtd8	1/1	Running	0	1h

「NAME」列にはポッド名が、「READY」列には起動しているコンテナ数が、「STATUS」列にはポッドの状態が、「RESTARTS」列には異常停止等の原因による再起動の回数が、「AGE」列にはデプロイしてからの経過時間が表示されます。

b. Web コンソールを使用する場合

b-1. Web コンソールにアクセスし、任意のプロジェクトを開きます。左メニューの「Applications」から「Pods」を選択し、作成したコンテナが存在すること、および Status を確認します。



DockerFile にて、EXPORSE 分を用いて作成したコンテナイメージか、"-p"オプションを指定して起動したことのあるコンテナイメージを利用した場合 OpenShift は自動で”service”を作成します。

”service”は OpenShift の Web コンソールから確認できます。

”service”確認画面の「create route」から”Route”を作成し、外部からのアクセスを出来るように設定します。

4章. 注意・補足事項

- OpenShift の環境では、root ユーザでの起動ができないため、“ftuser”というユーザを作成します。このユーザ名を変更したい場合、以下のスクリプトの該当箇所を編集してください。

ファイル名	編集箇所
esmpo_start.sh	2 行目 : FTUSER=ftuser
ft_daemon	2 行目 : FTUSER=ftuser

上記箇所の“ftuser”(小文字)の部分を任意のユーザ名に置き換えてください。

- Docker コマンドで作成したコンテナイメージと、OpenShift で作成したコンテナイメージには互換性があります。既にコンテナイメージをお持ちの場合 docker save / load コマンド等を利用して展開していただくことが可能です。※OpenShift で動作させるためには FT の起動スクリプトの編集が必要です。
- FT では、プロセス間通信識別子に特定の範囲を割り当て、固定することが可能です。しかし、OpenShift の環境で FT を利用する場合、このプロセス間通信識別子の割り当て、固定する設定は利用できません。

- DockerFile の実行内容

■DockerFile 内容は次の通りです。

```
FROM baseimage:latest .....①
RUN /bin/chmod 777 /etc/passwd.....②
ADD ["FT53RED","/tmp/FT53RED"] .....③
ADD ["ftsetup.sh","/tmp/ftsetup.sh"] .....④
RUN /bin/chmod +x /tmp/ftsetup.sh .....⑤
ADD ["ft_daemon","/tmp/ft_daemon"] .....⑥
RUN /bin/chmod +x /tmp/ft_daemon .....⑦
ADD ["t_ft_daemon","/tmp/t_ft_daemon"] .....⑧
RUN /bin/chmod +x /tmp/t_ft_daemon .....⑨
ADD ["esmpo_start.sh","/tmp/esmpo_start.sh"].....⑩
RUN /bin/chmod +x /tmp/esmpo_start.sh .....⑪
ADD ["yumadd.txt","/tmp/yumadd.txt"] .....⑫
RUN /tmp/ftsetup.sh.....⑬
RUN /tmp/esmpo_start.sh .....⑭
CMD /opt/SS/FileTransfer/bin/ft_daemon start .....⑮
```

それぞれの動作については次の通りです。

- ① コンテナを動作させるための OS のベースイメージを指定します。“baseimage”の部分は実際に使用するベースイメージに合わせて置き換えてください。
- ② OpenShift 環境ではランダムに割り当てられたユーザ ID でコンテナが起動されます。そのユーザ ID にユーザ名を設定するために、「/etc/passwd」を編集する必要があるため権限を付加します。
- ③ FT のインストール媒体「FT53RED」をコンテナの「/tmp」ディレクトリに格納します。
- ④ FT をインストールするためのスクリプトをコンテナの「/tmp」ディレクトリに格納します。
- ⑤ 格納したスクリプト「ftsetup.sh」に実行権を付加します。
- ⑥ 一般ユーザで起動できるように変更した FT の起動/停止スクリプト「ft_daemon」をコンテナの「/tmp」ディレクトリに格納します。
- ⑦ 格納したスクリプト「ft_daemon」に実行権を付加します。
- ⑧ ft_serv をフォアグラウンドで実行するように変更した FT の起動/停止スクリプト「t_ft_daemon」をコンテナの「/tmp」ディレクトリに格納します。
- ⑨ 格納したスクリプト「t_ft_daemon」に実行権を付加します。
- ⑩ FT を一般ユーザで起動するための設定を実施するスクリプトをコンテナの「/tmp」ディレクトリに格納します。
- ⑪ 格納したスクリプト「esmpo_start.sh」に実行権を付加します。
- ⑫ yum 実行時のプロキシサーバ情報ファイルをコンテナの「/tmp」ディレクトリに格納します。
- ⑬ 格納したスクリプト「ft_setup.sh」スクリプトを実行します。
- ⑭ 格納したスクリプト「esmpo_start.sh」スクリプトを実行します。
- ⑮ コンテナ起動時には「/opt/SS/FileTransfer/bin/ft_daemon start」するようという指示です。

- esmpro_start.sh の内容

DockerFile で実行するスクリプト esmpro_start.sh の内容は次の通りです。

■ esmpro_start.sh

```
#!/bin/bash
FTUSER=ftuser
temp=`/usr/bin/whoami`

if [ $temp ] ; then

    pid=`/bin/ps -e | /bin/egrep 'TimerProc' | /bin/sed -e 's/^  */' -e 's/ .*//'`
    if [ "${pid}" == "" ] ; then
        /opt/SS/DeliveryManager/common/bin/TimerProc > /dev/null &
    fi

    /bin/sleep 5

    /opt/SS/DeliveryManager/utility/regtool/regedit -setval esm/bs username ${FTUSER} STRING

    pid=`/bin/ps -e | /bin/egrep 'TimerProc' | /bin/sed -e 's/^  */' -e 's/ .*//'`
    if [ "${pid}" != "" ] ; then
        /bin/kill -9 ${pid}
    fi

fi
```

スクリプトの処理内容は次の通りです。

- コマンド「/bin/whoami」が失敗する場合（OpenShift 環境の場合）のみ、次の処理を実施します。
- TimerProc プロセスを起動します。
- regedit コマンドを使用し、一般ユーザ（デフォルトでは「ftuser」）をレジストリ値「username」に設定します。FT を一般ユーザで起動する設定です。
- TimerProc プロセスを停止します。

- **ftsetup.sh** の内容

DockerFile で実行するスクリプト **ftsetup.sh** の内容は次の通りです。

■ **ftsetup.sh** 内容

```
#!/bin/bash

/bin/cat /etc/yum.conf /tmp/yumadd.txt > /tmp/yumnew.txt
/bin/mv /etc/yum.conf /etc/yum.conf.org
/bin/cp /tmp/yumnew.txt /etc/yum.conf
/bin/yum -y update
/bin/yum -y install glibc.i686 libgcc.i686 libstdc++.i686
/bin/rpm -i /tmp/FT53RED
/bin/mkdir -m 777 /opt/SS/FileTransfer/bin/AUTO
/bin/mv /opt/SS/FileTransfer/bin/t_ft_daemon /opt/SS/FileTransfer/bin/t_ft_daemon.org
/bin/cp /tmp/t_ft_daemon /opt/SS/FileTransfer/bin/t_ft_daemon
/bin/chmod +x /opt/SS/FileTransfer/bin/t_ft_daemon
/bin/mv /opt/SS/FileTransfer/bin/ft_daemon /opt/SS/FileTransfer/bin/ft_daemon.org
/bin/cp /tmp/ft_daemon /opt/SS/FileTransfer/bin/ft_daemon
/bin/chmod +x /opt/SS/FileTransfer/bin/ft_daemon
```

スクリプトの処理内容は次の通りです。

- yum 実行時のプロキシサーバ情報（yumadd.txt の内容）を **/etc/yum.conf** に追記します。
- 「yum update」を実行し、コンテナのモジュールの最新化します。
- FT が依存するパッケージのインストールを実施します。
- FT のインストールを実施します。
- 一般ユーザで FT を運用するために、必要なディレクトリのアクセス権の設定をします。
- FT のインストール時に配置された、FT の起動/停止スクリプト「ft_daemon」「t_ft_daemon」を、コンテナでの運用に対応したスクリプトへ置き換えます。