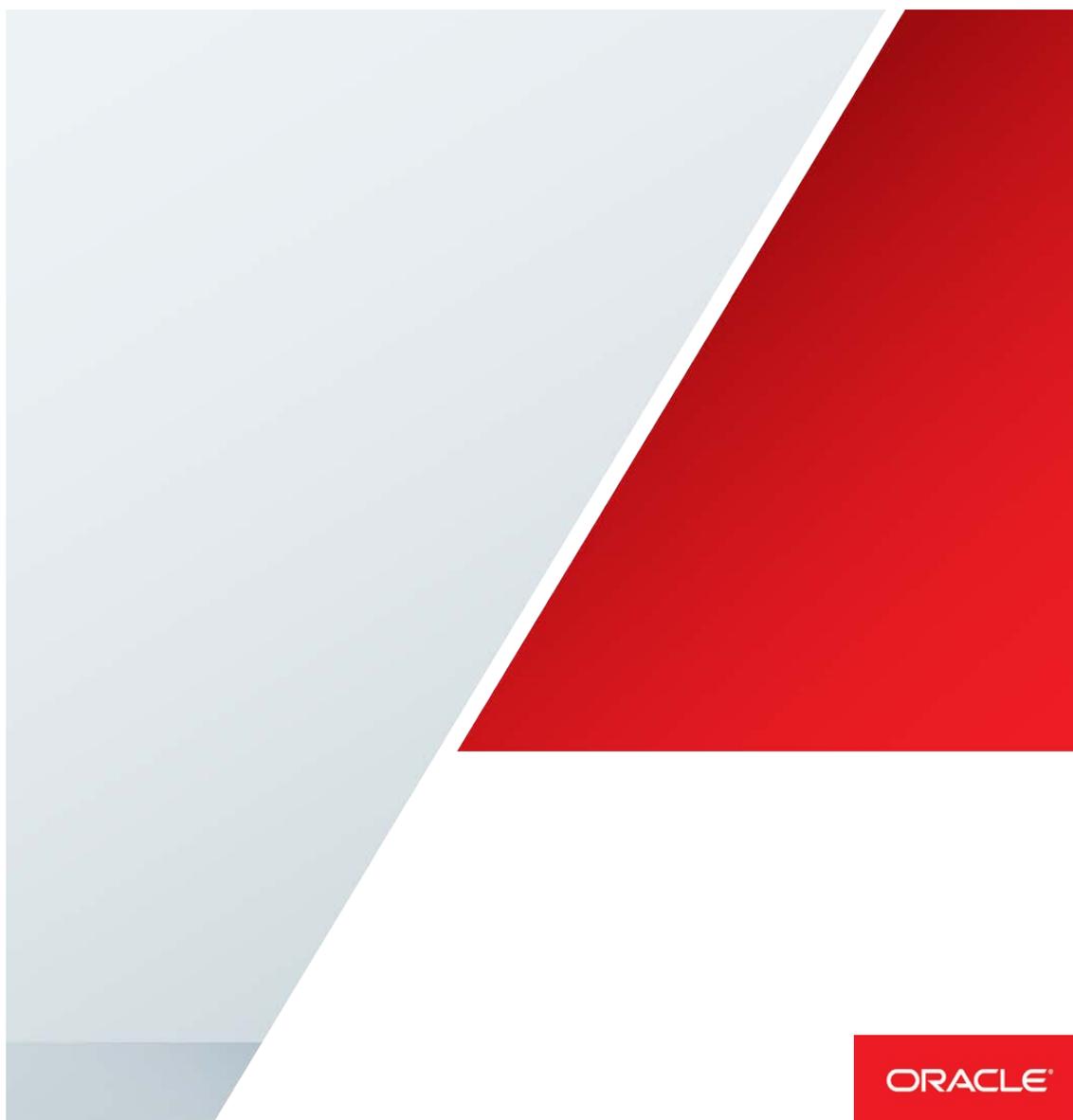


DockerコンテナでのOracle Tuxedoの実行



DockerでのOracle Tuxedoの実行

このたび、Docker コンテナでの Oracle Tuxedo の動作が保証されました。この保証の一環として、Oracle Tuxedo のイメージの構築に利用できる Dockerfile とサンプル・サポート・スクリプトの公開が、[クラスタ](#)・モード用と [SHM/非クラスタ](#)・モード用の両方について GitHub で開始されています。これらのイメージは、既存の Oracle Linux イメージの拡張版として構築されています。これらのスクリプトを使用して独自のスクリプトを作成することができます。

[Docker](#) は、分散アプリケーションを構築、パッケージ化、移動、および実行できるプラットフォームです。Docker のユーザーは、アプリケーションとその依存先となるすべてのライブラリおよびファイルをパッケージ化し、1 つの Docker イメージとしてまとめます。Docker イメージは、さまざまな Linux 環境に配布できる移植可能なアーティファクトです。配布されたイメージはコンテナのインスタンス化に使用でき、そのインスタンス化されたコンテナ上では、同じホスト・オペレーティング・システム上の他のコンテナで稼働しているアプリケーションとは独立してアプリケーションを実行させることができます。

以下の表に、2 つの Oracle Tuxedo バージョンの動作保証状況を示します。独自の Docker イメージを構築する場合は、Oracle Tuxedo、Linux、および Docker のこれらのバージョンを組み合わせることができます。

Oracle Tuxedo のバージョン	ホスト OS	カーネルのバージョン	Docker のバージョン
12.1.3.0.0	Oracle Linux 7	Unbreakable Enterprise Kernel Release 3 (3.8.13 以降)	1.3.9 以降
12.2.2.0.0	Oracle Linux 7	Unbreakable Enterprise Kernel Release 3 (3.8.13 以降)	1.3.9 以降

Oracle Tuxedo Dockerイメージ

オラクルは、Oracle Tuxedo Docker イメージの構築に利用できる Dockerfile とサポート・スクリプトを GitHub で公開しています。これらのイメージは、既存の Oracle Linux イメージ 6 に Oracle Tuxedo 12cR2 (12.2.2.0.0) のインストールを付け加えた拡張版として構築されています。

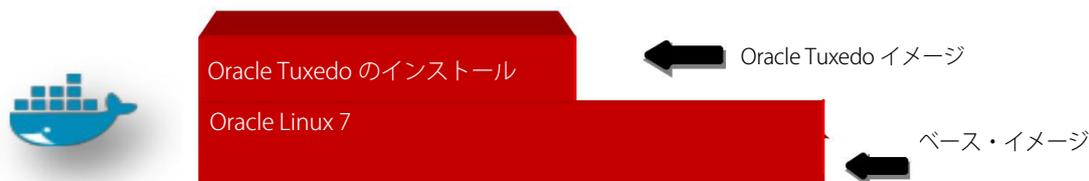


図1：Oracle Tuxedo Dockerイメージ

カスタム・ビルドのOracle Tuxedoイメージ

ユーザーは独自の Oracle Tuxedo Docker イメージを作成できます。作成を容易にするためにオラクルが作成した Dockerfile とスクリプトが [GitHub の Oracle Tuxedo の Dockerfile](#) で公開されていますので、これらをサンプルとして使用してください。

カスタム Oracle Tuxedo イメージを構築するための前提条件は次のとおりです。

1. Oracle Linux ベース・イメージ
2. GitHub で公開されている Dockerfile とスクリプト
3. Oracle Tuxedo のインストーラ

GitHubで公開されているDockerfileとスクリプト

Dockerfile

Oracle Tuxedo のインストール・イメージを作成する Dockerfile の機能は次のとおりです。

1. Oracle Linux ベース・イメージの拡張
2. サイレント・モードによる Oracle Tuxedo のインストール
3. 3 ノードで構成される Oracle Tuxedo サンプル・ドメインのセットアップ

スクリプト

スクリプトは Oracle Tuxedo イメージの作成を支援するもので、Oracle Linux ベース・イメージを拡張して Oracle Tuxedo ドメインの構成を追加するサンプルとして機能します。

DockerコンテナでのOracle Tuxedoのクラスタ化

これらの Docker コンテナを使用して、クラスタ構成または非クラスタ構成の Oracle Tuxedo ドメインを作成できます。ドメイン内で稼働する各サーバーはそれぞれの専用 Docker コンテナ内で動作し、必要に応じて同じホスト上の他のサーバーと通信することができます。

このトポロジの利点

- 従来型のデプロイメントに適している。
- Oracle Tuxedo ドメイン・イメージからコンテナをデプロイしやすい。
- クラスタのスケール・アップ/ダウンが容易である。
- 開発者に適している。

構築と実行の方法

Oracle Tuxedo のインストール・イメージの構築と、このイメージを拡張して Oracle Tuxedo ドメインを作成するのに必要な Dockerfile とサポート・スクリプトが、[GitHub](#) にあります。

Oracle Tuxedo イメージを構築してコンテナを起動するには、[クラスタ](#)と[非クラスタ](#)のディレクトリ構造全体をダウンロードしてください。

Dockerfile

/TuxedoUsers/Tuxedo-samples/tree/master/dockermpl サブディレクトリと/TuxedoUsers/Tuxedo-samples/tree/master/docker サブディレクトリに Oracle Tuxedo 12cR2 の Dockerfile が 1 つずつあります。1 つは Oracle Tuxedo のインストール・イメージを構築するための Dockerfile で、もう 1 つは 3 つのノードで構成される Oracle Tuxedo サンプル・ドメインを構築するための Dockerfile です。

スクリプト – 非クラスタ

サブディレクトリ/TuxedoUsers/Tuxedo-samples/tree/master/docker に tuxedo_docker.zip をダウンロードします。

Dockerfile - Oracle Tuxedo インストールを使用して Oracle Tuxedo イメージを構築します。

Dockerfile.template – 最終的な Dockerfile の生成に使用されるテンプレートです。

README.md – サンプルの実行手順です。

build.sh – Dockerfile、レスポンス・ファイル、および関連するビルド・スクリプトを生成するスクリプトです。

tuxedo_docker.zip – Docker に Oracle Tuxedo を構築するのに必要なすべてのファイルが含まれています。

install_tuxedo12.1.3.rsp – Oracle Tuxedo のサイレント・インストールに必要なレスポンス・ファイルです。

使用方法

1. 空のディレクトリで次の手順を実行します。
 - i. Oracle Tuxedo 12.1.3 Linux 64 ビット・インストーラを OTN からダウンロードします。
 - ii. `tuxedo_docker.zip` をこの Github ディレクトリからダウンロードします。
 - iii. (オプション) 最新の Oracle Tuxedo ローリング・パッチを My Oracle Support からダウンロードします。
2. `tuxedo_docker.zip` を解凍します。
3. `build.sh` を実行します。

`oracle/tuxedo` とタグ付けされた Docker イメージができています。

次に、`docker run -i -t oracle/tuxedo /bin/bash` を実行して新しいコンテナ内でこのイメージを起動できます。これは、`bash` プロンプトを使用してコンテナ内に入るコマンドです。新しいコンテナをテストする場合は、空のディレクトリで `simpapp_runme.sh` を実行します。すると、Oracle Tuxedo の `simpapp` アプリケーションが作成され、実行されます。

スクリプト - クラスタ

サブディレクトリ `TuxedoUsers/Tuxedo-samples/tree/master/dockemp` にすべてのファイルをダウンロードします。

build.sh - 環境をチェックし、サイレント・モード用レスポンス・ファイルや Dockerfile を生成します。

Dockerfile.template - 最終的な Dockerfile の生成に使用されるテンプレートです。

fix_locations.sh - ローカル環境に適合するようにインストーラのスクリプトとレスポンス・ファイルを変更するスクリプトです。

README.md - サンプルの実行手順です。

simpapp_runme.sh - Oracle Tuedo のサンプル・アプリケーション `simpapp` を作成して実行するシェル・スクリプトです。

start_tlisten.sh - ホストのネットワーク・アドレスの指定されたポートで `tlisten` を起動するスクリプトです。

tuxedo.docker - シェル・スクリプトとテンプレートのパラメータ・リストと定義です。

tuxedo12.1.3.rsp.template – Oracle Tuxedo のサイレント・インストール・モードで使用するレスポンス・ファイルのテンプレートです。

tuxedo12.1.3_silent_install.sh.template - tuxedo12.1.3_silent_install.sh のテンプレート・ファイルです。

使用方法

1. まず、simpapp アプリケーションを含む Oracle Tuxedo Docker イメージを作成する docker プロジェクト non-Clustered を実行します。
2. ディレクトリからファイルを取得します。
3. コンテナを構築します (docker-compose build)。
4. コンテナを起動します (docker-compose up -d)。
5. この時点で、3 つの Docker コンテナ (dockerm_node1_1、dockerm_node2_1、dockerm_node3_1) が稼働しているはずですが。
6. dockerm_node1_1 でシェルを開きます (docker exec -it dockerm_node1_1 /bin/bash)。
7. setenv.sh スクリプトを source で実行します (source setenv.sh)。
8. simpappmp_runme.sh スクリプトを実行します (sh ./simpappmp_runme.sh)。
9. これで、Oracle Tuxedo の 3 ノード・クラスターができ、それぞれのノードで simpserve のコピーが 1 つずつ稼働しているはずですが。
10. 複数のノードでロードが実行されたことを確認するには、次のコマンドを実行します。
 - tmadmin
 - d site1
 - psc
 - d site2
 - psc
 - d site3
 - psq
 - q

11. 1つのノードでクライアント・リクエストの一部が処理され、別のノードで他の一部が処理されたことを確認できるはずです。

Oracle TuxedoをDockerで実行する場合のその他の考慮事項

Docker コンテナが再起動されて IP アドレスが変更されると、その Docker コンテナ内で稼働する Oracle Tuxedo のアドレスが新しいものになります。コンテナの再起動前にそのサーバーと通信していたアプリケーションや他のサーバーは、その通信が不可能になります。コンテナの再起動後に IP アドレスが変更される問題を解決するには、Docker 上に DNS サーバーを構成し、Oracle Tuxedo ドメインで DNS 名が使用されるように構成します。

Oracle Tuxedo の構成、サーバー・ログ、ファイル・ストアなどは、すべてコンテナのファイル・システムに保管されます。Docker コンテナが破損すると、ファイル・システム全体を失うことになります。この問題への対策としては次の2つの方法があります。

- ホストのファイル・システムを使用してコンテナのローカル・ファイル・システムを保存する。
- ドメインのファイル・システムを保管するための"データ専用"コンテナを維持する。

ファイル・システムへの依存性を最小限に抑えるために、次の対策を行うことを推奨します。

- TLog、Queue ストアなどのデータストアをデータベースで管理する。

Oracle Tuxedo のインストール・イメージで作成された Oracle Tuxedo イメージへのパッチ適用やアップグレードを行うには、次の手順を実行します。

1. Oracle Tuxedo インストールの Docker イメージを拡張する方法で、アップグレードまたはパッチ適用を行います。
2. Docker cp (copy) コマンドを使用して、ホストまたは"データ専用"コンテナのいずれかのコピー先にドメイン・フォルダをコピーします。
3. コンテナを削除します。
4. アップグレードまたはパッチ適用によって拡張されたイメージから新しいコンテナを実行します。
5. Docker cp (copy) コマンドを使用して、アップグレードされたコンテナにドメイン・フォルダをコピーします。

Docker および Linux のコンテナには、セキュリティに関する次の懸念事項があります。

- 別々のコンテナで実行中のコードを互いに分離できるかどうか懸念されます。現時点では、そのような環境での Oracle Tuxedo の運用に影響を及ぼす既知の問題はありません。
- Docker イメージの提供元に関するセキュリティ上の懸念点もあります。信頼できる提供元のみから Docker イメージを入手する必要があり、更新頻度や、Docker Hub での管理の特性について理解しておく必要があります。
- Docker および Linux テクノロジーの最新情報を常に把握し、それぞれについて発生してい

るセキュリティの問題を常に認識しておく必要があります。

- Docker コンテナのデフォルト・ネットワーク・モードである"ブリッジ・ネットワーク"では、マルチキャストがサポートされません。Docker コンテナの"ホスト・ネットワーク"ではマルチキャストがサポートされますが、ホストのネットワーク・スタックが利用されるため、独立性は低下します。Docker コンテナで実行する場合は、Oracle Tuxedo クラスターリング・プロトコルとしてユニキャストを使用することを推奨します。

結論

アーティファクトに移植性があり、複数の Linux 環境に容易に配布できるため、Docker は運用の簡素化とコスト削減を実現するうえで有望なテクノロジーです。オラクルは、高まりつつあるお客様の関心に応えるために、Docker コンテナ内での Oracle Tuxedo の動作を保証し、Docker コンテナで稼働する Oracle Tuxedo 構成の作成をサポートする Dockerfile とスクリプトを用意しました。オラクルは、これらがお客様のお役に立つことを願い、今後も Docker 環境のサポート範囲の充実を目指していきます。