

詳細

リフトアンドシフト方式によるクラウド移行のための開発者ガイド

目次

1 はじめに	1
2 RED HAT APPLICATION MIGRATION TOOLKIT の使用による移行の単純化	2
2.1 Java EE アプリケーションの分析.....	2
2.2 レポートの理解.....	6
2.3 問題の修正.....	7
2.4 ビルドの再テスト	8
3 RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM でのプロジェクト実行	9
3.1 Red Hat JBoss Enterprise Application Platform のインストール.....	10
3.2 WildFly Maven プラグイン	10
3.3 アプリケーションの構成.....	10
3.4 アプリケーションのデプロイ.....	10
3.5 アプリケーションのテスト.....	10
3.6 アプリケーションのシャットダウン	10
4 RED HAT OPENSIFT CONTAINER PLATFORM へのモノリスのデプロイ	10
4.1 Red Hat OpenShift プロファイルの追加.....	11
4.2 Red Hat OpenShift プロジェクトの作成.....	11
4.3 モノリスのデプロイ	14
4.4 バイナリービルドを使用したアプリケーションのデプロイ	14
5 まとめ	16



fb.com/RedHatJapan
twitter.com/RedHatJapan
linkedin.com/company/red-hat

jp.redhat.com

はじめに

クラウドモデルの浸透に伴い、クラウドネイティブ開発に移行する方法を模索する組織が増えていきます。マイクロサービス、自律的な開発チーム、アジャイルで継続的なデプロイメント、そして、コンテナ化とオーケストレーションを使用したクラウドデプロイメントを活用すれば、クラウドモデルとサービスのみを使用して新しいアプリケーションを開発できます。必要な時間やコストを考えると、すべてのレガシー・アプリケーションを完全にリライトすることは、残念ながらほぼ実現不可能です。

組織が俊敏性と競争力を維持していくためには、既存の Java™ アプリケーションの移行およびモダナイズと、クラウドネイティブ開発の導入が必須です。ほとんどの企業において、このプロセスは、既存の機能とデータを可能な限り再利用すること、既存のワークロードを先進的なデプロイメント・プラットフォームに移行すること、そして最終的に新しいプロセス、製品、およびテクノロジーを適用することから構成されます。

「リフトアンドシフト」方式のモダナイゼーションは、クラウドネイティブ開発への魅力的な第一歩であり、次のことを実施します。

- 既存のモノリシックなワークロードをコンテナ化する
- ワークロードを PaaS (Platform-as-a-Service) にデプロイする
- 外部の統合とデータをレガシー・プラットフォーム上に残す

これらのステップが完了すると、開発者はモノリスを抑制し、厳選したアプリ内機能を少しずつマイクロサービスに置き換える作業を開始できます。

このガイドでは、Red Hat® ソフトウェアを使用したリフトアンドシフト方式のプロセス (図 1) について説明します。この移行プロセスでは次のことを実施します。

1. **Red Hat Application Migration Toolkit** を使用して既存のモノリシック・アプリケーションを分析し、オープンで標準的なインタフェースに移行する
2. **Red Hat JBoss® Enterprise Application Platform** で実行できるよう、コードと構成を更新する
3. 自動スケーリング、統合されたクラスタリング、フェイルオーバーなどの機能でアプリケーションを強化できるよう、モノリスを **Red Hat OpenShift® Container Platform** 上にコンテナとしてデプロイする

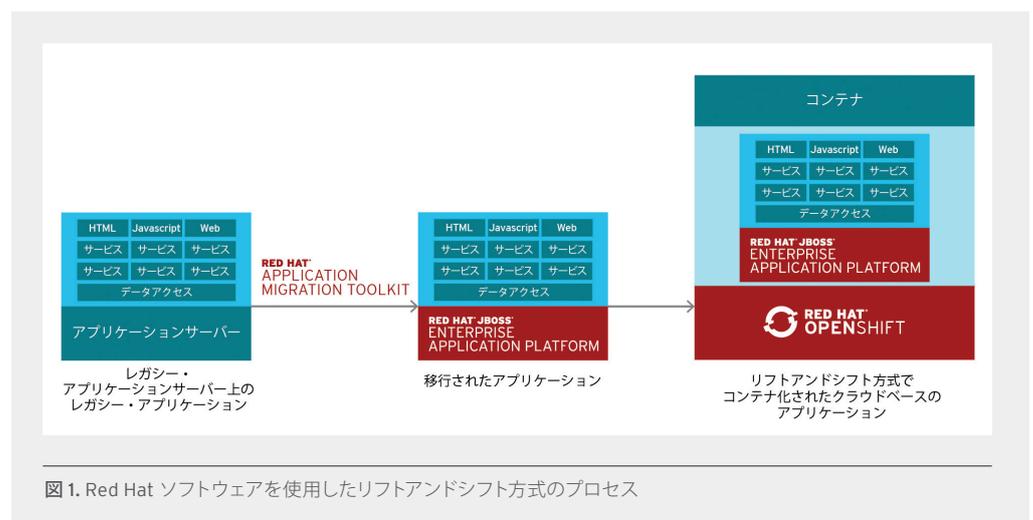


図 1. Red Hat ソフトウェアを使用したリフトアンドシフト方式のプロセス

アプリケーションを Red Hat OpenShift Container Platform にデプロイした後、追加のツールを使用して、機能を個別にマイクロサービスに変換していくことができます。このプロセスで機能を細かく分割し、並列化と自律性を高めることができるので、次のようなツールを使って価値実現までの時間の短縮に注力できます。

- [Thorntail / MicroProfile](#)
- [Spring Boot](#)
- [Node.js](#)
- [Eclipse Vert.x](#)

RED HAT APPLICATION MIGRATION TOOLKIT の使用による移行の単純化

Red Hat Application Migration Toolkit は拡張およびカスタマイズが可能なルールベースのツールで、Java アプリケーションの移行を単純化するのに役立ち、次のような目的で使用されています。

- 計画と作業見積もり
- 移行の問題の特定および解決策の提案
- 詳細レポートの評価
- 組み込み型のルールと移行パスの使用
- ルールの拡張とカスタマイズ性の提供
- ソースコードまたはアプリケーション・アーカイブの分析

Red Hat Application Migration Toolkit は共通のリソースを探し、テクノロジーと、アプリケーションを移行する際の既知の問題点を明らかにします。このツールキットを使用するとアプリケーションで使用されるテクノロジーの概要を確認し、詳細なレポートを取得できるので、Java EE アプリケーションを Red Hat JBoss Enterprise Application Platform に移行する作業の見積もり、文書化、および移行の実施に役立てることができます。

注：Red Hat Application Migration Toolkit は通常、アプリケーションの移行およびモダナイゼーションのためのプログラム全体の一部にすぎません。このようなプログラムは通常、明確に定義された反復可能なフェーズに従って数週間から数カ月をかけて実施されるものであり、組織内の多くの人々が関与します。

Red Hat の哲学と実証済みの方法論についての詳細は、[Red Hat Application Migration Toolkit のドキュメント](#)と[開発者向けホームページ](#)をご覧ください。

JAVA EE アプリケーションの分析

Red Hat Application Migration Toolkit は、プロジェクトのソースディレクトリやアプリケーション・アーカイブなど、アプリケーションのアーティファクトを分析します。その後生成される HTML レポートを参照すると、変更が必要な領域を確認できます。このツールキットは、以前のバージョンの Red Hat JBoss Enterprise Application Platform (EAP) や、Oracle WebLogic Server や IBM® WebSphere Application Server などの他のアプリケーションサーバー・プラットフォームから Java アプリケーションを移行するために使うことができます。

Red Hat Application Migration Toolkit は、複数の方法でインストールおよび使用できます。

- **Web コンソール:** Red Hat Application Migration Toolkit の Web コンソールを使用すると、開発者がチームで多数のアプリケーションの移行とモダナイゼーションの取り組みを評価し、優先順位を付けることができます。アプリケーションをプロジェクトごとにグループ分けして分析し、さまざまなレポートを生成して結果を確認できます。
- **コマンドライン・インタフェース (CLI):** CLI は、開発者がアプリケーションの移行とモダナイゼーションの取り組みを評価し、優先順位を付けることを可能にするコマンドラインツールで、さまざまなレポートを生成して分析結果を確認できます。シンプルに使えるので、単一アプリケーションの分析に最適です。

- **Eclipse プラグイン**：Red Hat Application Migration Toolkit の Eclipse プラグインは、移行やモダナイゼーションの取り組みで実際に変更を加える開発者が Eclipse と Red Hat Developer Studio から直接機能を利用できるようにします。Red Hat Application Migration Toolkit を使用してプロジェクトを分析し、ソースコード内に存在する移行の問題のマーキング、問題解決のためのガイダンスの提供、そして、可能な場合はコードの自動置き換えを行います。

これらのアクセス方法はすべて、[Red Hat Application Migration Toolkit](#) ページからダウンロードできます。インストール手順については、各アクセス方法のドキュメントを参照してください。複数のユーザーで使用する場合は、Web コンソールが適しています。説明をシンプルにするため、この資料では CLI を使って説明します。以下の手順は、Red Hat Application Migration Toolkit の操作を要約したものです。

1. RED HAT APPLICATION MIGRATION TOOLKIT CLI を検証する

Red Hat Application Migration Toolkit CLI をインストールしたら、次のコマンドを実行してツールが正しくインストールされたことを確認します。

```
$ ${HOME}/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli --version
```

次のような応答が表示されます。

```
Using RHAMT at /root/rhamt-cli-4.0.0.Beta4
```

```
> Red Hat Application Migration Toolkit (RHAMT) CLI, version 4.0.0.Beta4.
```

2. プロジェクトに対して RED HAT APPLICATION MIGRATION TOOLKIT CLI を実行する

Red Hat Application Migration Toolkit CLI は、さまざまなオプションを使用して実行方法をコントロールできます。ここでは、「monolith」というプロジェクトに対して CLI でツールキットを実行し、同じ名前でレポートを生成します。下記の `~/projects/monolith` を実際のプロジェクト名に置き換えてください。

```
$ ~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli \
--sourcemode \
--input ~/projects/monolith \
--output ~/rhamt-reports/monolith \
--overwrite \
--source weblogic \
--target eap:7 \
--packages com.redhat weblogic
```

--source オプションと --target オプションが使われていることに注意してください。これらのオプションを使うと、Red Hat Application Migration Toolkit でサポートされている移行パスを指定してターゲットにできます。移行パスには次のものが含まれます。

- Oracle WebLogic Server
- IBM WebSphere Application Server
- Red Hat JBoss EAP 5/6/7

ツールキットの実行が完了するのを待ちます。完了すると、次のような応答が表示されます。

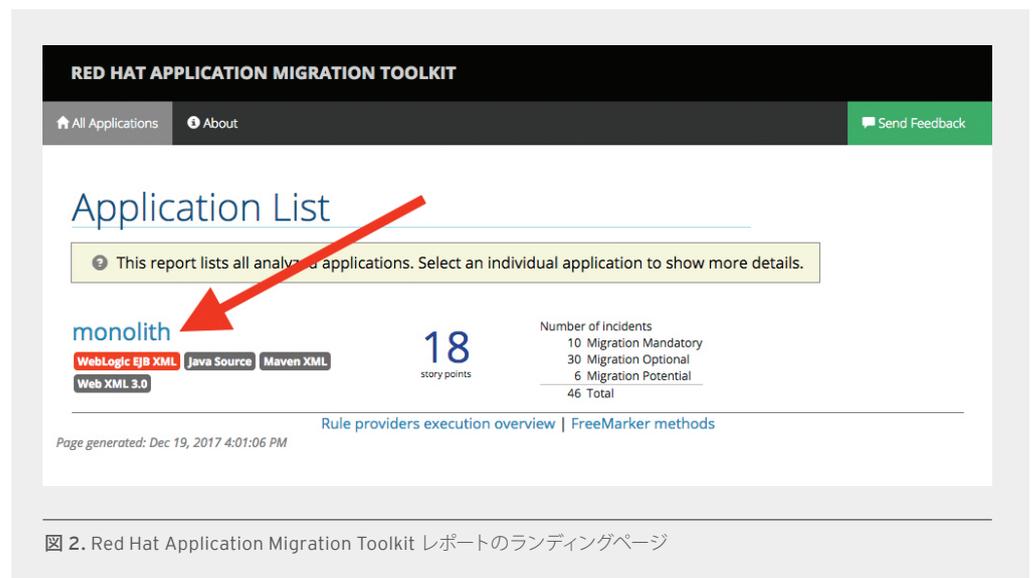
```
Report created: ~/rhamt-reports/monolith/index.html
```

3. 結果を確認する

レポートを表示するには、ローカルブラウザで次のファイルにアクセスします。

~/rhant-reports/monolith/index.html

レポートのランディングページ (図 2) が表示されます。メインレポートのランディングページには、処理されたアプリケーションと、発生した問題が一覧表示されます。ページの各行には、ストーリーポイント、インシデント数、およびアプリケーションで使用されているテクノロジーのたまかな概要が表示されます。



The screenshot shows the 'RED HAT APPLICATION MIGRATION TOOLKIT' interface. At the top, there are navigation links for 'All Applications' and 'About', and a 'Send Feedback' button. The main heading is 'Application List'. Below this, a message states: 'This report lists all analyzed applications. Select an individual application to show more details.' A red arrow points to the 'monolith' application entry. The entry includes the application name 'monolith', a list of technologies (WebLogic EJB XML, Java Source, Maven XML, Web XML 3.0), a large number '18' representing story points, and a table of incident counts: 10 Migration Mandatory, 30 Migration Optional, 6 Migration Potential, and 46 Total. At the bottom, there are links for 'Rule providers execution overview' and 'FreeMarker methods', and a footer note: 'Page generated: Dec 19, 2017 4:01:06 PM'.

図 2. Red Hat Application Migration Toolkit レポートのランディングページ

図 2 の `monolith` リンクをクリックすると、プロジェクトのダッシュボード (図 3) に遷移します。

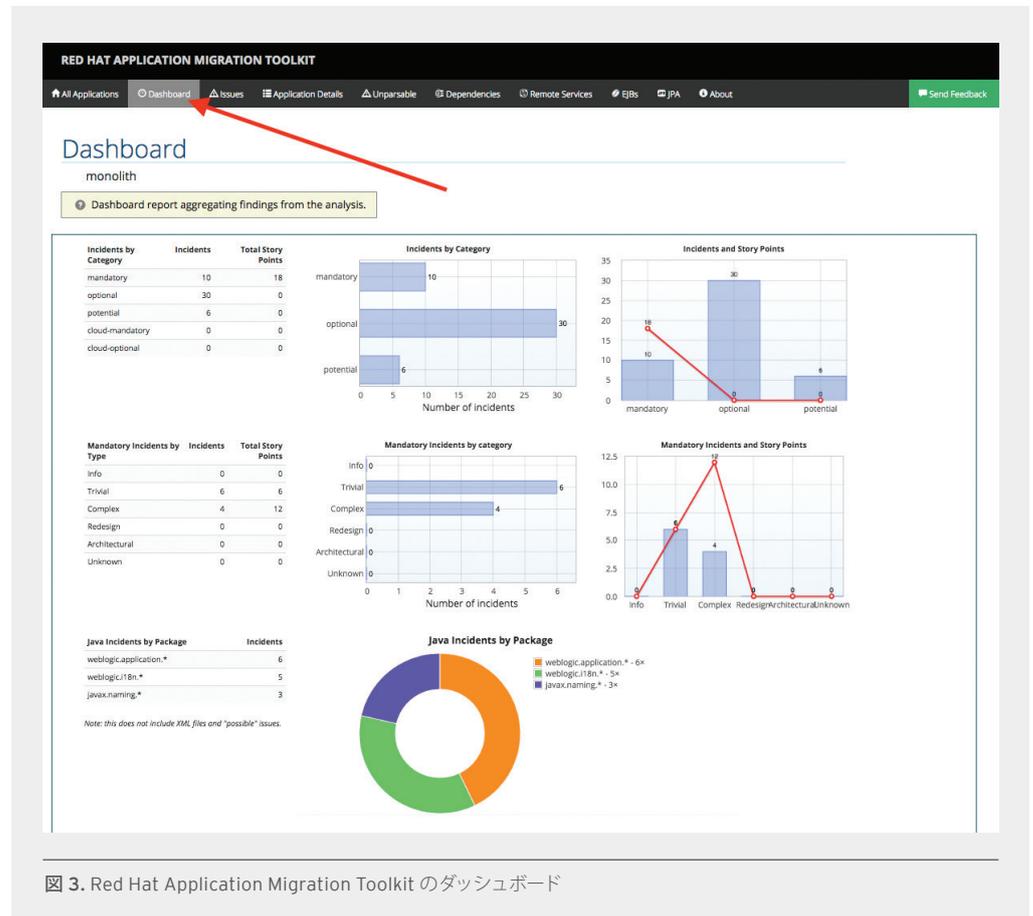


図 3. Red Hat Application Migration Toolkit のダッシュボード

レポートの理解

ダッシュボードには、アプリケーション移行作業全体の概要が表示されます。内容は次のとおりです。

- カテゴリ別のインシデントとストーリーポイント
- 提案された変更の作業レベル別のインシデントとストーリーポイント
- パッケージ別のインシデント

注：ストーリーポイントはアジャイルソフトウェア開発で一般に使用される抽象的なメトリックで、機能の実装や変更に必要な作業の相対的なレベルを評価します。Red Hat Application Migration Toolkit では、アプリケーションの個々の構成要素やアプリケーション全体の移行に必要な作業レベルを表すのにストーリーポイントを使用しています。作業レベルは、移行するアプリケーションのサイズと複雑さによって大きく異なります。

メインページにもいくつかのサブページがあり、上部のメニューからアクセスできます。

- **All Applications** : スキャンされた全アプリケーションのリストを表示します。
- **Dashboard** : 特定のアプリケーションの概要を示します。
- **Issues** : 注意を必要とする問題がすべてまとめられています。

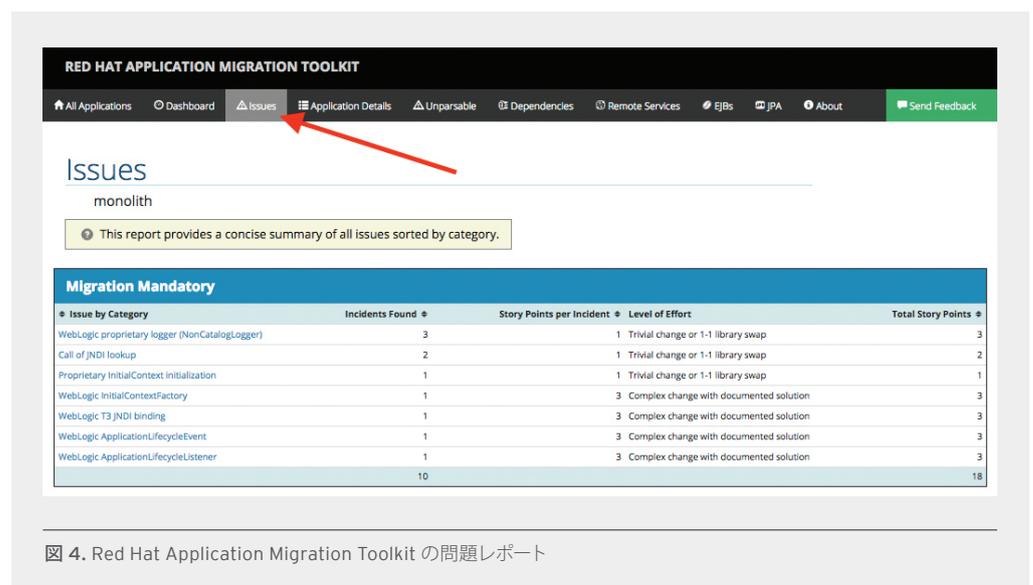
- **Application Details** : 移行作業中に注意を要する可能性がある、アプリケーション内のすべてのリソースの詳しい概要が表示されます。
- **Unparsable** : Red Hat Application Migration Toolkit が予測した形式で解析できなかったすべてのファイルを表示します。例えば、拡張子が .xml または .wsdl のファイルは XML ファイルとみなされません。XML パーサーで解析できなかった場合は、このセクションと、個々のファイルをリストにしたセクションで問題が報告されます。
- **Dependencies** : アプリケーション内で検出されたすべての Java パッケージの依存関係を表示します。
- **Remote Services** : アプリケーション内で検出されたすべてのリモートサービスへの参照を表示します。
- **EJBs** : アプリケーション内で検出された Enterprise Java Beans のリストを表示します。
- **JBPM** : 分析中に発見された Java Business Process Management 関連のリソースをすべて一覧表示します。
- **JPA** : アプリケーションで検出されたすべての Java Persistence API (アプリケーション・プログラミング・インタフェース) 関連のリソースについての詳細を表示します。
- **About** : Red Hat Application Migration Toolkit の現在のバージョンと、さらなる支援が必要なときに役立つリンクを表示します。

注: 上記のセクションのいくつかは、プロジェクトで実際に検出された問題によっては表示されないことがあります。

Red Hat Application Migration Toolkit レポートを取得したら、次はアプリの移行を始めます。

問題の修正

図 4 の Issues Report タブには、対象のアプリケーションに関するすべての既知の問題について、より詳細なリストが表示されます。ユーザーが問題をより深く理解するのに役立つリンクを一覧表示し、移行とモダナイゼーションのガイダンスを提供します。



The screenshot shows the 'Issues' report for a 'monolith' application. The navigation bar includes 'All Applications', 'Dashboard', 'Issues', 'Application Details', 'Unparsable', 'Dependencies', 'Remote Services', 'EJBs', 'JPA', 'About', and 'Send Feedback'. The 'Issues' section contains a summary: 'This report provides a concise summary of all issues sorted by category.' Below this is a table titled 'Migration Mandatory' with the following data:

Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
WebLogic proprietary logger (NonCatalogLogger)	3	1	Trivial change or 1-1 library swap	3
Call of JNDI lookup	2	1	Trivial change or 1-1 library swap	2
Proprietary InitialContext initialization	1	1	Trivial change or 1-1 library swap	1
WebLogic InitialContextFactory	1	3	Complex change with documented solution	3
WebLogic T3 JNDI binding	1	3	Complex change with documented solution	3
WebLogic ApplicationLifecycleEvent	1	3	Complex change with documented solution	3
WebLogic ApplicationLifecycleListener	1	3	Complex change with documented solution	3
	10			18

図 4. Red Hat Application Migration Toolkit の問題レポート

多くのアプリケーションには次のようなプラットフォーム固有のコードがあり、標準の Java EE インタフェースを使用するためには更新する必要があります。

- **起動コード**: アプリケーションの起動時や停止時に機能を実行したりジョブをスケジュールしたりするのに使用する
- **ロギングコード**: プラットフォーム固有のロギング機能を使用する
- **メッセージングコード**: 旧式またはプラットフォーム固有のメカニズムを使用する

問題レポートは、それぞれの問題について取りうる解決策を提案します。実際には、これらの問題の対処には移行の一環として追加の検討が必要になることがあります。これらの変更を実装するとコードの可搬性が大幅に向上します。

必要な変更を加えたら、Maven でアプリケーションのビルドとパッケージ化を行い、更新後のコードを問題なくコンパイルできることを確認します。

```
$ mvn clean package
```

コードが正常にビルドされたら (BUILD SUCCESS と表示)、問題レポートの次の問題を修正します。

ビルドの再テスト

すべての問題を処理したら、Red Hat Application Migration Toolkit を再度実行して、正常に移行できたことを確認します。

1. プロジェクトに対して RED HAT APPLICATION MIGRATION TOOLKIT CLI を実行する

以下のコマンドを実行して古いビルドアーティファクトをクリーンアップし、Red Hat Application Migration Toolkit CLI を再実行して新しいプロジェクトを分析します。先ほどと同様、~/projects/monolith は実際のプロジェクトの場所に置き換えてください。

```
$ mvn clean && \  
~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli \  
--sourceMode \  
--input ~/projects/monolith \  
--output ~/rhamt-reports/monolith \  
--overwrite \  
--source weblogic \  
--target eap:7 \  
--packages com.redhat weblogic
```

プロセスが完了すると、CLI に次のように表示されます。

```
Report created: /root/rhamt-reports/monolith/index.html
```

2. 結果を確認する

次の URL でレポートのローカル Web ページを再読み込みします。

```
~/rhamt-reports/monolith/index.html
```

移行が成功したら、対応するアプリケーションのストーリーポイントはゼロ (0) になります。これは、アプリケーションが正常にモダナイズされ、Red Hat JBoss EAP に移行する準備が整ったことを示します。

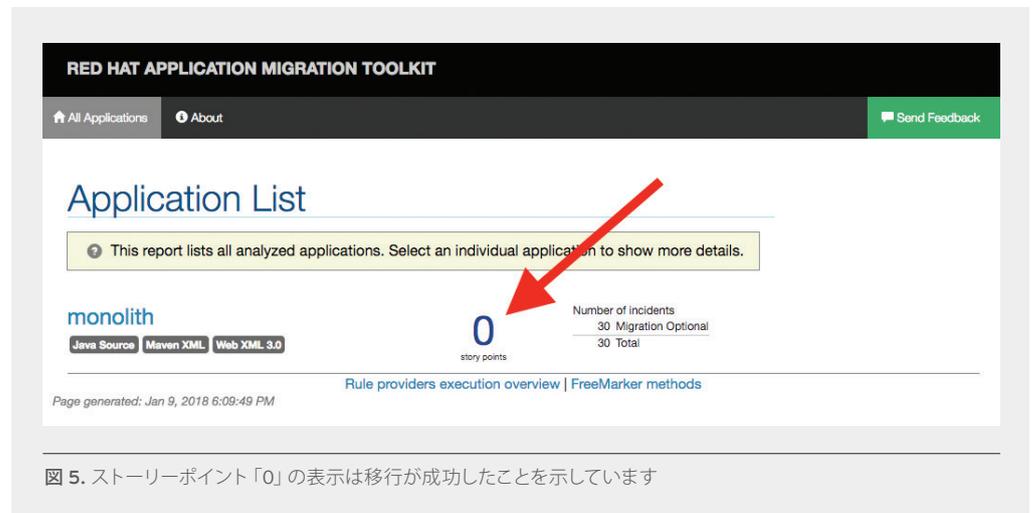


図 5. ストーリーポイント「0」の表示は移行が成功したことを示しています

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM でのプロジェクト実行

より標準的な Java EE インタフェースを使用するようにアプリケーションを更新したので、次はデプロイしてテストし、Red Hat JBoss EAP が提供する機能をいくつか使ってみます。

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM のインストール

Red Hat JBoss EAP の最新リリース版をダウンロードします。次にターミナルウィンドウで以下のコマンドを実行して Red Hat JBoss EAP をローカルにインストールします。ファイル名は実際にダウンロードしたバージョンに合わせて変更してください。

```
$ unzip -d $HOME $HOME/jboss-eap-7.2.0.zip
```

次に、JBOSS_HOME 環境変数を設定します。

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0
```

Red Hat JBoss EAP のインストールは以上の簡単な手順で終了です。

WILDFLY MAVEN プラグイン

Red Hat JBoss EAP には、Apache Maven から直接プラットフォームの停止、起動、デプロイ、設定を行うことができる [wildfly-maven-plugin](#) ツールがあります。この Maven プラグインツールの設定が終了したら、ツールを使って以下のことができます。

- フル Java EE と Java EE Web プロファイルの切り替え
- データベースリソースの設定
- Java Message Service のトピックなどのサービスの設定

これらの機能の設定はすべて `pom.xml` ファイルで行います。

アプリケーションの構成

Red Hat Application Migration Toolkit の使用後、アプリケーションはほぼ標準ベースになるため、次のように、Red Hat JBoss EAP を起動してアプリケーションの起動、リソースの追加、シャットダウンを行うことで構成できます。

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; \ mvn wildfly:start  
wildfly:add-resource wildfly:shutdown
```

この手順が正常に終了すると、`BUILD SUCCESS` というメッセージが表示されます。

注: `wildfly:start` および `wildfly:shutdown` を使用しているのは、実行中のアプリケーションサーバーがなければ `add-resource` コマンドを使えないためです。このリソースを追加した後は、このコマンドを再度実行する必要はありません。

アプリケーションのデプロイ

アプリケーションをデプロイする準備が整ったので、次のコマンドでデプロイします。

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; mvn wildfly:run
```

サーバーが起動すると、次のように表示されます。

```
Deployed "ROOT.war" (runtime-name: "ROOT.war")
```

アプリケーションのテスト

ブラウザで次の URL を入力してアプリケーションにアクセスします。

```
http://localhost:8080
```

警告があれば、コンソール出力に表示されます。

アプリケーションのシャットダウン

次の作業を始める前に、ターミナルウィンドウで `CTRL+C` を押してアプリケーションを終了してください。

RED HAT OPENSIFT CONTAINER PLATFORM へのモノリスのデプロイ

アプリケーションを Red Hat JBoss EAP で起動および実行できたら、新たに更新したそのモノリスを Red Hat OpenShift Container Platform に移動できます。以下の手順は、Red Hat OpenShift Container Platform クラスタへのアクセスを前提としています。クラスタにアクセスできない場合は、ローカルに [Red Hat OpenShift Container Platform をインストール](#) することができます。

RED HAT OPENSIFT プロファイルの追加

pom.xml ファイルを開き、Red Hat OpenShift プロファイルを追加します。

```
<profile>
  <id>openshift</id>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.6</version>
        <configuration>
          <webResources>
            <resource>
              <directory>${basedir}/src/main/webapp/WEB-INF</directory>
              <filtering>true</filtering>
              <targetPath>WEB-INF</targetPath>
            </resource>
          </webResources>
          <outputDirectory>deployments</outputDirectory>
          <warName>ROOT</warName>
        </configuration>
      </plugin>
    </plugins>
  </build>
</profile>
```

図 6. Red Hat OpenShift プロファイル (pom.xml ファイル)

RED HAT OPENSIFT プロジェクトの作成

まず、Red Hat OpenShift Container Platform の Web コンソールを開きます。

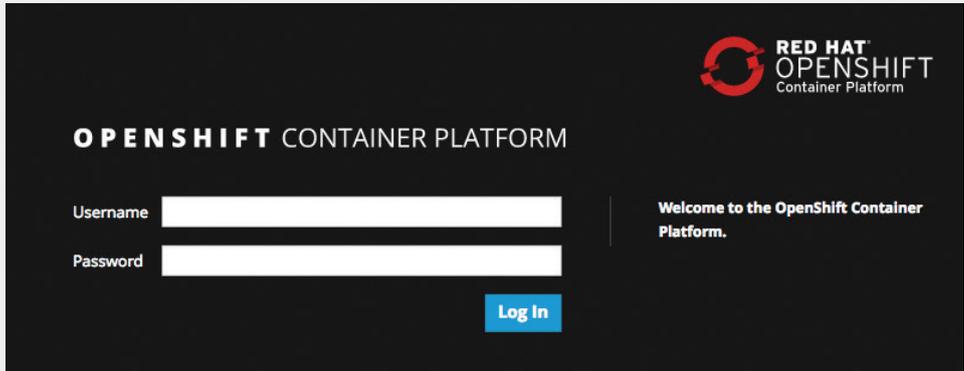


図 7. Red Hat OpenShift Container Platform の Web コンソール

Red Hat OpenShift の資格情報でログインします。

- ユーザー名: XXXXXX
- パスワード: XXXXXX

Red Hat OpenShift Container Platform のランディングページが表示されます。

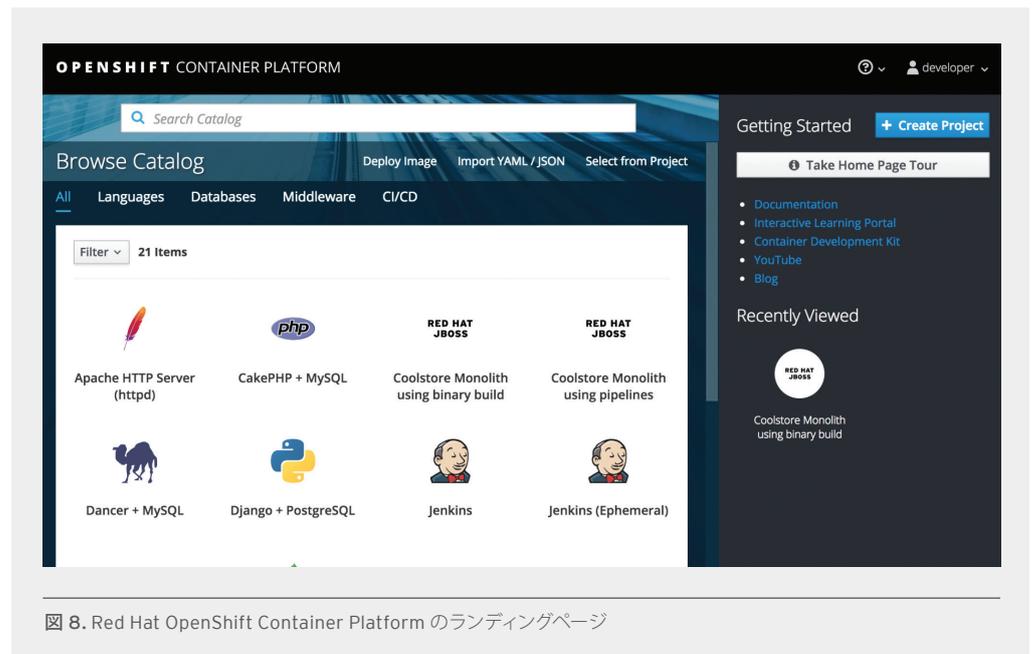
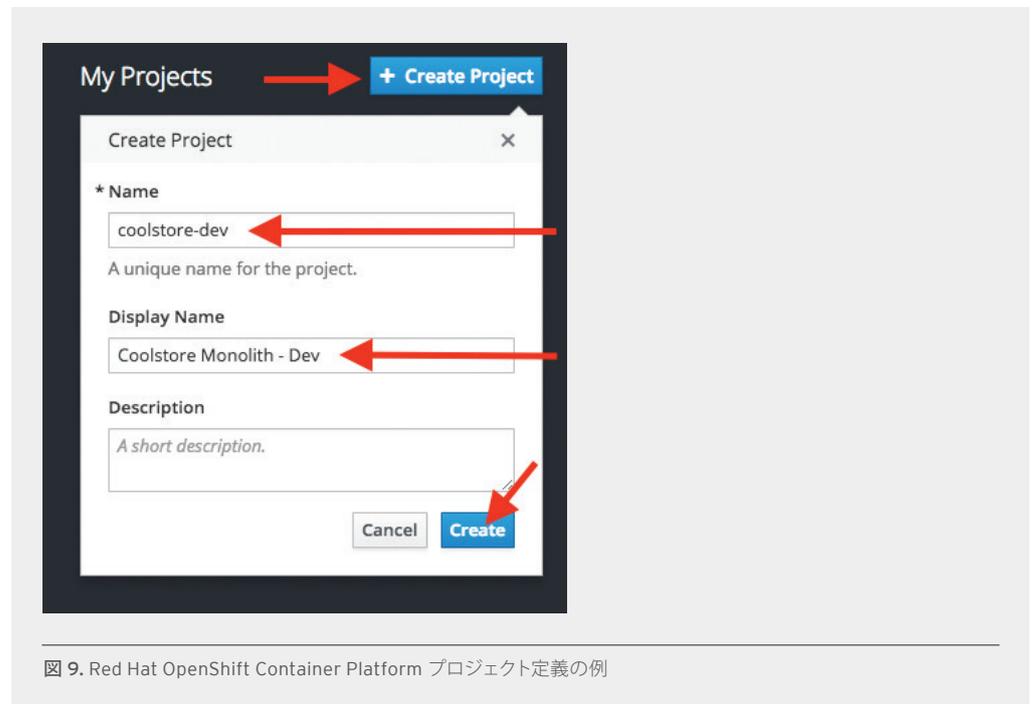


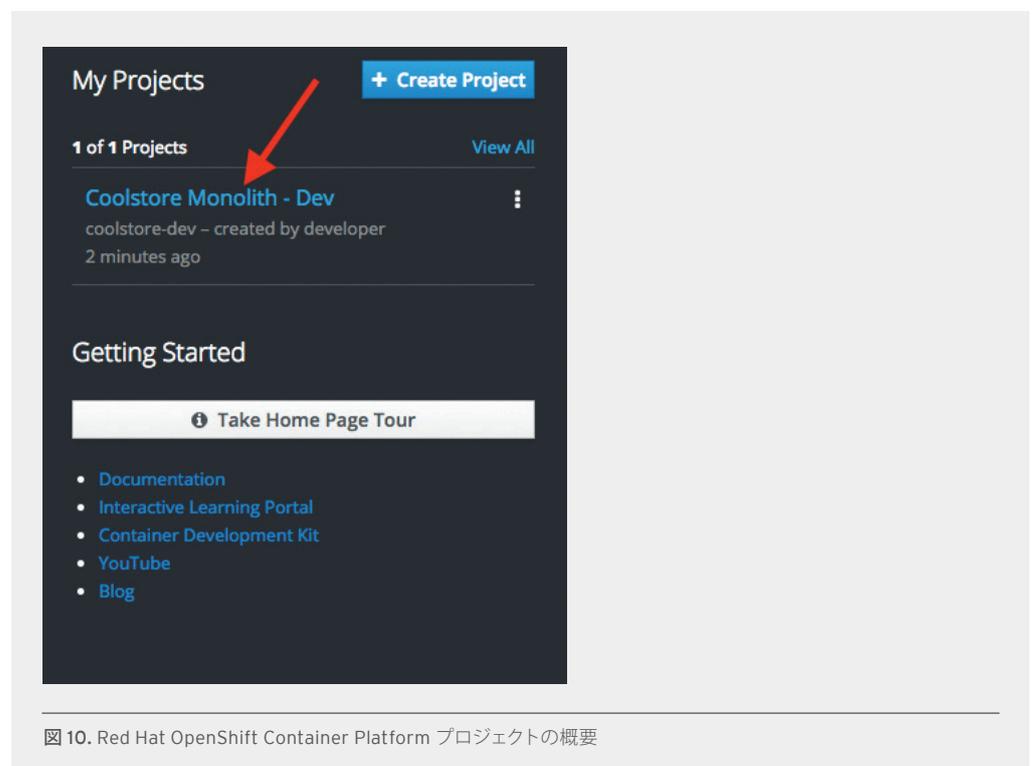
図 8. Red Hat OpenShift Container Platform のランディングページ

Create Project をクリックしてフィールドに入力し、Create をクリックします。この例では、アプリケーション名は「coolstore」です。

- Name: coolstore-dev
- Display Name: Coolstore Monolith - Dev
- Description: 空欄のまま



新しく作成したプロジェクトの名前をクリックします。



プロジェクトの概要画面に移動します。まだ何もありませんが、これから追加します。

モノリスのデプロイ

CLI を使用してモノリスのコンポーネントをデプロイします。CLI でモノリステンプレートをデプロイするには、以下のコマンドを実行します。

まず、先ほど作成した開発者プロジェクトに切り替えます。

```
$ oc project coolstore-dev
```

テンプレートをデプロイします。

```
$ oc new-app coolstore-monolith-binary-build
```

この手順を実行すると Red Hat JBoss EAP だけでなくアプリケーションに必要なデータベースもデプロイされますが、アプリケーションのビルドは開始されません。

次に、以下の URL からモノリス概要ページを開き、モノリステンプレート項目が作成されていることを確認します。

```
https://$OPENSHIFT_MASTER/console/project/coolstore-dev/
```

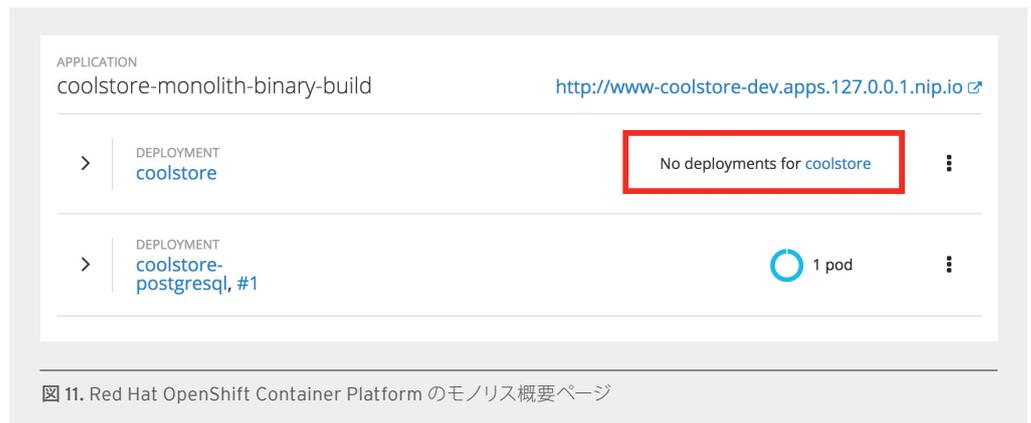


図 11. Red Hat OpenShift Container Platform のモノリス概要ページ

プロジェクト概要を見るとコンポーネントがデプロイされていることがわかりますが、「No deployments for coolstore」と表示されていることに注意してください。前の手順で作成したコンテナイメージはまだデプロイしていませんが、次の手順で行います。

バイナリービルドを使用したアプリケーションのデプロイ

この開発プロジェクトでは、バイナリービルドと呼ばれるプロセスを使用します。パブリック Git リポジトリをポイントして S2I (source-to-image) ビルドプロセスでダウンロード、ビルド、およびコンテナイメージの作成を行うこともできますが、バイナリービルドのプロセスでは、ローカルでビルドしてアーティファクト (.war ファイルなど) をアップロードするだけです。このバイナリー・デプロイメントにより、ビルドプロセスを大幅に迅速化できます。

まず、openshift Maven プロファイルを使用してプロジェクトをもう一度ビルドします。これにより、Red Hat OpenShift Container Platform での使用に適したバイナリーが作成されます (これはまだコンテナイメージではなく、単なる .war ファイルです)。このプロジェクトをこれから oc コマンドでビルドします。

プロジェクトをビルドします。

```
$ mvn clean package -Popenshift
```

ビルドが完了し、BUILD SUCCESS メッセージが表示されるのを待ちます。

最後に、.war ファイルを取得して Red Hat JBoss EAP と結合し、Linux® コンテナイメージを生成するビルドプロセスを開始します。テンプレートで作成される DeploymentConfig オブジェクトにより、このイメージは自動でプロジェクトにデプロイされます。

```
$ oc start-build coolstore --from-file=deployments/ROOT.war
```

Red Hat OpenShift Web コンソールを見ると、アプリケーションがビルド中であることが確認できます。この例では、PostgreSQL データベースがすでにデプロイ済みです。



図 12. モノリス概要ページ。ビルドは進行中ですが、デプロイメントはありません

ビルドとデプロイが完了するのを待ちます。

```
$ oc rollout status -w dc/coolstore
```

このコマンドは、デプロイメントの完了を待つためによく使われます。使用する際は、成功のメッセージが返されることを確認してください。処理が終わると以下のメッセージが表示されます。

```
replication controller "coolstore-1" successfully rolled out.
```

注:上記のコマンドでサーバーからのエラー (ServerTimeout) が報告された場合は、成功が報告されるまでコマンドを再実行してください。

プロセスが完了すると、データベースとモノリスに青い丸印が表示され、アプリケーションが正常にデプロイされたことがわかります。



図 13. アプリケーションが正常にデプロイされると青い丸印が表示されます

詳細 リフトアンドシフト方式によるクラウド移行のための開発者ガイド

Route リンクをクリックして以下の URL にアクセスし、アプリケーションをテストします。

`http://www-coolstore-dev.$ROUTE_SUFFIX` : この URL にアクセスすると、Red Hat OpenShift Container Platform 上で実行されている coolstore モノリスがブラウザで開きます。

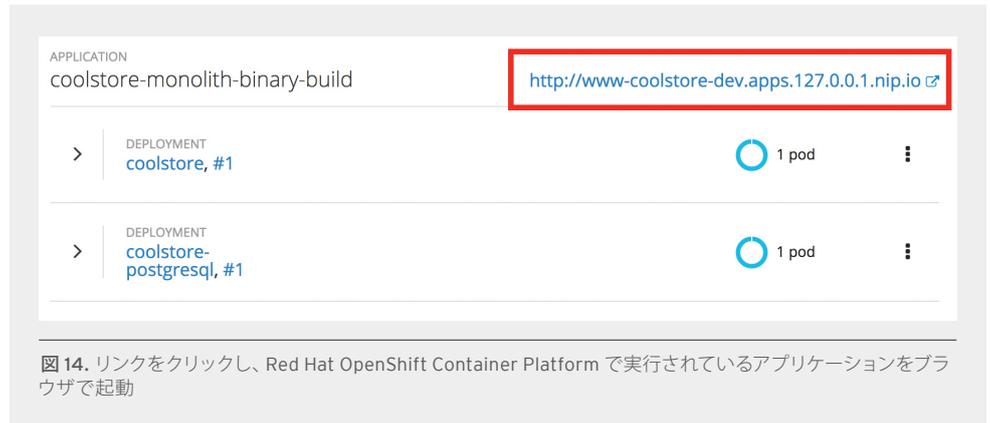
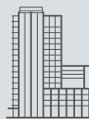


図 14. リンクをクリックし、Red Hat OpenShift Container Platform で実行されているアプリケーションをブラウザで起動

まとめ

このプロセスでは、Red Hat JBoss Enterprise Application Platform と Red Hat OpenShift Container Platform を使用して、既存の Java EE アプリケーションをクラウドに移行する方法を説明しました。これにより、最新のテクノロジーを使用して、分散およびコンテナ化された環境でアプリケーションを適切に実行できるようにすると同時に、モノリスを段階的に小さなマイクロサービスに分割することによって、アプリケーションをモダナイズする準備を整えることができます。



RED HAT について

エンタープライズ・オープンソース・ソフトウェア・ソリューションのプロバイダーとして世界をリードする Red Hat は、コミュニティとの協業により高い信頼性と性能を備える Linux、ハイブリッドクラウド、コンテナ、および Kubernetes テクノロジーを提供しています。Red Hat は、新規および既存 IT アプリケーションの統合、クラウドネイティブ・アプリケーションの開発、Red Hat が提供する業界トップレベルのオペレーティングシステムへの標準化、複雑な環境の自動化、セキュリティ保護、運用管理を支援します。受賞歴のあるサポート、トレーニング、コンサルティングサービスを提供する Red Hat は、Fortune 500 企業に信頼されるアドバイザーです。クラウドプロバイダー、システムインテグレーター、アプリケーションベンダー、お客様、オープンソース・コミュニティの戦略的パートナーとして、Red Hat はデジタル化が進む将来に備える企業を支援します。

アジア太平洋 +65 6490 4200 apac@redhat.com	インドネシア 001 803 440 224	マレーシア 1 800 812 678	中国 800 810 2100
オーストラリア 1 800 733 428	日本 0120 266 086 03 5798 8510	ニュージーランド 0800 450 503	香港 800 901 222
インド +91 22 3987 8888	韓国 080 708 0880	シンガポール 800 448 1430	台湾 0800 666 052



fb.com/RedHatJapan
twitter.com/RedHatJapan
linkedin.com/company/red-hat