



Red Hat build of Quarkus 1.3

Red Hat OpenShift Container Platform での
Quarkus アプリケーションのデプロイ

Red Hat build of Quarkus 1.3 Red Hat OpenShift Container Platform での Quarkus アプリケーションのデプロイ

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat OpenShift Container Platform に Quarkus アプリケーションをデプロイする方法を説明します。

目次

前書き	3
第1章 RED HAT ビルドの QUARKUS	4
第2章 RED HAT OPENSIFT CONTAINER PLATFORM	5
第3章 QUARKUS OPENSIFT エクステンションを使用した OPENSIFT への QUARKUS アプリケーションのデ プロイ	6
第4章 S2I を使用した OPENSIFT での QUARKUS アプリケーションのデプロイ	8

前書き

アプリケーション開発者は、1つのビルドコマンドで Apache Maven および **quarkus-openshift** エクステンションを使用して、または従来の Source-to-Image (S2I) 手法を使用して、Red Hat OpenShift Container Platform に Quarkus アプリケーションをデプロイすることができます。どちらの手法を使用しても、作成されるイメージは完全にサポートされます。開発サポートの対象となるビルドプロセスおよびツールの詳細は、[開発サポートの対象範囲](#) のページを参照してください。

前提条件

- OpenJDK 11 がインストールされ、**JAVA_HOME** 環境変数が Java SDK の場所を指定していること。Red Hat ビルドの Open JDK は、Red Hat カスタマーポータル[の Software Downloads](#) ページから入手可能です (ログインが必要です)。
- Apache Maven 3.6.2 以降がインストールされていること。Maven は [Apache Maven Project](#) の Web サイトから入手できます。
- Quarkus Maven プロジェクトがあること。Maven を使用した簡単な Quarkus アプリケーションのビルドに関する説明は、[『Red Hat ビルドの Quarkus のスタートガイド』](#) を参照してください。



注記

Quarkus Maven プロジェクトの完全なサンプルについては、[Quarkus quickstart archive](#) をダウンロードするか、**Quarkus Quickstarts** Git リポジトリをクローンしてください。この例は **getting-started** ディレクトリにあります。

- Red Hat OpenShift Container Platform クラスターにアクセスでき、最新バージョンの OpenShift CLI (oc) がインストールされていること。oc のインストールに関する詳細は、[『OpenShift Container Platform クラスターのインストールおよび設定』](#) ガイドを参照してください。

第1章 RED HAT ビルドの QUARKUS

Red Hat ビルドの Quarkus は、コンテナおよび Red Hat OpenShift Container Platform と使用するために最適化された Kubernetes ネイティブ Java スタックです。Quarkus は、Eclipse MicroProfile、Apache Kafka、RESTEasy (JAX-RS)、Hibernate ORM (JPA)、Spring、Infinispan、Apache Camel などの一般的な Java 標準、フレームワーク、およびライブラリーと連携するように設計されています。

Quarkus のディペンデンシーインジェクション (依存性の注入) ソリューションは、CDI (コンテキストとディペンデンシーインジェクション) をベースとし、エクステンションフレームワークを備えているので、機能の拡張、およびフレームワークの設定、起動、アプリケーションへの統合が可能です。

Quarkus は、コンテナファーストという手法で Java アプリケーションをビルドします。この手法により、Java で書かれたマイクロサービスベースのアプリケーションのビルドが大幅に容易になるほか、これらのアプリケーションがサーバーレスコンピューティングフレームワークで実行している関数を呼び出すことができるようになります。これにより、Quarkus アプリケーションのメモリーフットプリントは小さくなり、起動時間は高速化されます。

第2章 RED HAT OPENSIFT CONTAINER PLATFORM

OpenShift Container Platform は、コンテナ化されたアプリケーションを開発し、実行するための Kubernetes ベースのプラットフォームです。アプリケーションおよびアプリケーションをサポートするデータセンターが、わずかなシステムとアプリケーションから、数百万ものクライアントに対応する数千ものシステムへと拡張できるように設計されています。

OpenShift は、アプリケーションのコンテナイメージをビルドするための2つのワークフロー（ソースワークフローとバイナリーワークフロー）をサポートします。どちらのワークフローも Source-to-Image (S2I) 機能をベースとしており、両方のワークフローはソースワークフローを使用して S2I ビルドに依存します。主な違いは、ソースワークフローは OpenShift 内でアプリケーションのデプロイ可能なアーティファクトを生成し、バイナリーワークフローは OpenShift 外でこれらのバイナリーアーティファクトを生成する点です。いずれも OpenShift 内でアプリケーションコンテナイメージをビルドします。バイナリーワークフローは、コンテナイメージを生成する OpenShift S2I ビルドのソースとしてアプリケーションバイナリーを提供します。

第3章 QUARKUS OPENSIFT エクステンションを使用した OPENSIFT への QUARKUS アプリケーションのデプロイ

従来の Source-to-Image (S2I) ソースワークフローは、OpenShift 内でアプリケーションのデプロイ可能なアーティファクトを生成します。Quarkus OpenShift エクステンションは、S2I バイナリーワークフローを使用して、より効率的なデプロイメントプロセスを提供します。ソースからビルドする代わりに、エクステンションはローカルファイルシステムから JAR ファイルをアップロードします。その結果、ビルドプロセスは従来の S2I 手法よりも最大で 10 倍速くなります。ローカルで開発する場合、およびビルドサーバーまたは継続的インテグレーション (CI) システムから開発する場合に、Quarkus OpenShift エクステンションを使用して、ソースから繰り返し可能なビルドを実行することができます。



注記

Quarkus OpenShift エクステンションは、開発およびテストの目的でのみ使用してください。実稼働環境では、「[4章 S2I を使用した OpenShift での Quarkus アプリケーションのデプロイ](#)」で説明されている従来の S2I 手法の使用を検討してください。

手順

1. Quarkus Maven プロジェクトが含まれるディレクトリーに移動します。
2. OpenShift エクステンションを既存プロジェクトに追加するには、以下のコマンドを入力します。

```
./mvnw quarkus:add-extension -Dextensions="openshift"
```



注記

新規プロジェクトの作成時に **-Dextensions="openshift"** 引数を含めて、Quarkus OpenShift エクステンションを追加することができます。

OpenShift エクステンションを追加すると、スクリプトは以下の依存関係を **pom.xml** ファイルに追加します。

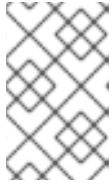
```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-openshift</artifactId>
</dependency>
```

3. 信頼されていない証明書を使用している場合は、**src/main/resources/application.properties** ファイルに以下の行を追加します。

```
quarkus.kubernetes-client.trust-certs=true
```

4. Open JDK 11 の Red Hat Enterprise Linux 7 イメージを使用するように OpenShift を設定するには、以下の行を **application.properties** ファイルに追加します。

```
quarkus.s2i.base-jvm-image=registry.access.redhat.com/openjdk/openjdk-11-rhel7
```



注記

IBM Z インフラストラクチャーにデプロイする場合は、**application.properties** ファイルに **quarkus.s2i.base-jvm-image=registry.access.redhat.com/openj9/openj9-11-rhel8** を追加します。

5. OpenShift ルートを作成するには、以下の行を **application.properties** ファイルに追加します。

```
quarkus.openshift.expose=true
```

6. 変更を **application.properties** ファイルに保存します。

7. OpenShift CLI (oc) にログインします。

```
oc login
```

8. 新しい OpenShift プロジェクトを作成するには、以下のコマンドを実行します。このコマンドの **PROJECT_NAME** は、新規プロジェクトの名前に置き換えます。

```
oc new-project PROJECT_NAME
```

9. プロジェクトを OpenShift にデプロイするには、Quarkus Maven プロジェクトディレクトリーに以下のコマンドを入力します。

```
./mvnw clean package -Dquarkus.kubernetes.deploy=true
```

10. OpenShift プロジェクトでデプロイされたすべてのアプリケーションの名前およびルートを表示するには、以下のコマンドを入力します。

```
oc get route
```

11. アプリケーションの完全な URL を表示するには、以下のコマンドを入力します。ここで、**APPLICATION_NAME** は OpenShift プロジェクトにデプロイされたアプリケーションの名前になります。

```
export URL="http://$(oc get route APPLICATION_NAME -o jsonpath='{.spec.host}')"  
echo "Application URL: $URL"  
curl $URL/hello
```

12. ルートの **hello** エンドポイントに HTTP 要求を作成するには、以下のコマンドを入力します。

```
curl $URL/hello
```

第4章 S2I を使用した OPENSIFT での QUARKUS アプリケーションのデプロイ

従来の Source-to-Image (S2I) 手法は、Red Hat OpenShift Container Platform でのアプリケーションのデプロイにおける推奨される方法として、広く使用されています。S2I では、Git リポジトリを使用するか、ビルド時にソースをアップロードして、ソースコードをビルドコンテナに提供する必要があります。この手法を使用して、実稼働環境に Quarkus アプリケーションをデプロイします。

前提条件

- Git リポジトリでホストされる Quarkus Maven プロジェクトがあること。

手順

1. Quarkus Maven プロジェクトが含まれるディレクトリーに移動します。
2. **pom.xml** ファイルと同じレベルで、**.s2i** という名前の隠しディレクトリーを作成します。
3. **.s2i** ディレクトリーに **environment** という名前のファイルを作成し、以下の内容を追加します。

```
ARTIFACT_COPY_ARGS=-p -r lib/ *-runner.jar
```

4. 変更をリモート Git リポジトリにコミットし、プッシュします。
5. OpenShift CLI (oc) にログインします。

```
oc login
```

6. 新しい OpenShift プロジェクトを作成するには、以下のコマンドを実行します。このコマンドの **PROJECT_NAME** は、新規プロジェクトの名前に置き換えます。

```
oc new-project PROJECT_NAME
```

7. サポートされる OpenShift イメージをインポートするには、以下のコマンドを入力します。

```
oc import-image --confirm openjdk/openjdk-11-rhel7 --  
from=registry.access.redhat.com/openjdk/openjdk-11-rhel7
```



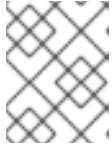
注記

IBM Z インフラストラクチャーにデプロイする場合は、**oc import-image --confirm openj9/openj9-11-rhel8 --from=registry.access.redhat.com/openj9/openj9-11-rhel8** を入力します。

このイメージの詳細は、[Red Hat OpenJDK 11 Java Applications](#) ページを参照してください。

8. OpenShift でプロジェクトをビルドするには、以下のコマンドを入力します。**GIT_PATH** は Quarkus プロジェクトをホストする Git リポジトリへのパス、**PROJECT_NAME** は作成した OpenShift プロジェクトに置き換えます。

```
oc new-app openjdk-11-rhel7 GIT_PATH --name=PROJECT_NAME
```



注記

IBM Z インフラストラクチャーにデプロイする場合は、**oc new-app openj9/openj9-11-rhel8 GIT_PATH --name=PROJECT_NAME** を入力します。

このコマンドを実行すると、プロジェクトのビルド、アプリケーションの作成、そして OpenShift サービスのデプロイが行われます。

9. OpenShift ルートを作成するには、以下のコマンドを入力します。

```
oc expose service/PROJECT_NAME
```

10. 新規ルートを表示するには、以下のコマンドを入力します。このコマンドの **APPLICATION_NAME** は、OpenShift プロジェクトにデプロイされたアプリケーションの名前に置き換えます。

```
export URL="http://$(oc get route APPLICATION_NAME -o jsonpath='{.spec.host}')"  
echo "Application URL: $URL"
```

11. ルートの **hello** エンドポイントに HTTP 要求を作成するには、以下のコマンドを入力します。

```
curl $URL/hello
```

12. アプリケーションを使用するには、前のコマンドで返された URL を Web ブラウザーで入力します。
13. プロジェクトの更新バージョンをデプロイするには、更新を Git リポジトリにプッシュしてから、以下のコマンドを入力します。

```
oc start-build PROJECT_NAME
```

14. ビルドの完了後にブラウザーページを更新し、変更を確認します。

改訂日時: 2020-11-18 01:18:31 UTC