Abstract

This Release Note contains important information related to Red Hat OpenShift Application Runtimes Spring Boot 2.1.x
# Table of Contents

PREFACE .............................................................................................................. 3

CHAPTER 1. REQUIRED INFRASTRUCTURE COMPONENT VERSIONS .......................... 4

CHAPTER 2. SUPPORTED SPRING BOOT RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS ..................................................... 5

CHAPTER 3. FEATURES ...................................................................................... 6
  3.1. NEW AND CHANGED FEATURES ............................................................... 6
  3.2. DEPRECATED FEATURES ......................................................................... 6
  3.3. TECHNOLOGY PREVIEW .......................................................................... 6

CHAPTER 4. RELEASE COMPONENTS .................................................................... 7
  4.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE ....................... 7
  4.2. TESTED AND VERIFIED ARTIFACTS INTRODUCED IN THIS RELEASE ........ 7
  4.3. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE .......... 7
  4.4. ARTIFACTS REMOVED IN THIS RELEASE .............................................. 7
  4.5. ARTIFACTS DEPRECATED IN THIS RELEASE .......................................... 7

CHAPTER 5. FIXED ISSUES .................................................................................. 8

CHAPTER 6. KNOWN SPRING BOOT ISSUES ....................................................... 9
  6.1. SB-379: MISSING APR/NATIVE LIBRARY IN THE OPENSHIFT-OPENJDK IMAGE 9
  6.2. CONNECTION BETWEEN A RHEL 8-BASED DATABASE APPLICATION AND A RHEL 7-BASED MYSQL 5.7 DATABASE FAILS DUE TO TLS PROTOCOL VERSION MISMATCH 9

CHAPTER 7. KNOWN ISSUES AFFECTING REQUIRED INFRASTRUCTURE COMPONENTS ................................................................. 10

CHAPTER 8. ADVISORIES RELATED TO THIS RELEASE ....................................... 12
# CHAPTER 1. REQUIRED INFRASTRUCTURE COMPONENT VERSIONS

The following versions of infrastructure components are required for all runtimes distributed as part of a RHOAR release. Red Hat does not provide support for components listed below, with the exception of components explicitly designated as supported.

<table>
<thead>
<tr>
<th>Component name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maven</td>
<td>3.5.0</td>
</tr>
<tr>
<td>Fabric8 Maven Plugin</td>
<td>4.2.0</td>
</tr>
<tr>
<td>JDK[a][b]</td>
<td>OpenJDK 8, OpenJDK 11[c]</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 7[d]</td>
<td>7.7</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 8[e]</td>
<td>8.0</td>
</tr>
<tr>
<td>OpenShift Container Platform (OCP)[f]</td>
<td>3.11</td>
</tr>
<tr>
<td>Minishift</td>
<td>1.34.1 or later</td>
</tr>
<tr>
<td>CDK[g]</td>
<td>3.8.0</td>
</tr>
<tr>
<td>git</td>
<td>2.0 or later</td>
</tr>
<tr>
<td>oc command line tool</td>
<td>3.11 or later[h]</td>
</tr>
</tbody>
</table>

[a] A full JDK installation is required, as JRE does not provide tools for compiling Java applications from source.

[b] Red Hat OpenJDK is supported by Red Hat

[c] OpenJDK 9 is not supported by Red Hat.

[d] For deploying RHOAR-based applications on stand-alone RHEL in a production environment.

[e] For deploying RHOAR-based applications on stand-alone RHEL in a production environment.

[f] OCP is supported by Red Hat

[g] CDK is supported by Red Hat

[h] The version of the `oc` CLI tool should correspond to the version of OCP that you are using.
CHAPTER 2. SUPPORTED SPRING BOOT RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

The following resource defines the supported configurations and integrations of Red Hat products with RHOAR Spring Boot:

- For a list of technologies that are supported for integration with RHOAR Spring Boot in production environments see the Supported Spring Boot configurations and integrations.

- For a list of RHOAR Spring Boot 2.1.x runtime artifacts and their versions see the Spring Boot 2.1.6 component details page.
CHAPTER 3. FEATURES

3.1. NEW AND CHANGED FEATURES

New features are provided as Technology Preview in this release.

3.2. DEPRECATED FEATURES

No features or functionalities are marked as deprecated in this release.

3.3. TECHNOLOGY PREVIEW

DeKorate

This release of RHOAR Spring Boot includes Dekorate, a Java annotation processor for Kubernetes, formerly developed under the name AP4K. Dekorate is a tool for automatically updating Kubernetes and OpenShift configuration files without the need to manually edit individual XML, YAML or JSON templates. When declared as a dependency in your Maven project, DeKorate automatically picks up annotations and changes to properties that you set in your application and automatically updates the corresponding deployment configuration and resource definition templates. Dekorate is provided as Technology Preview.

Vert.X Reactive Components

The RHOAR Spring Boot 2.1.6 introduces a set of supported Starters for designing reactive applications. The productized starters are based on community releases Spring WebFlux and Reactor Netty, with a set of additional Eclipse Vert.x extensions for the Spring Boot runtimes that extend the reactive capabilities of Spring WebFlux, to include an asynchronous IO API that handles network communication between individual application components. This enables you to create a fully Red Hat-supported reactive stack that you can use to build your Spring Boot applications. The set of Eclipse Vert.x reactive components for Spring Boot is provided as Technology Preview.
CHAPTER 4. RELEASE COMPONENTS

4.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE

No supported artifacts are introduced in this release.

4.2. TESTED AND VERIFIED ARTIFACTS INTRODUCED IN THIS RELEASE

NOTE

The following artifact has been reintroduced as an unsupported component of this release. The starter is tested and verified for use with the Spring Boot 2.1.6 release, but Red Hat does not provide support for using it in a production environment.

- org.apache.cxf

4.3. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE

- dev.snowdrop.vertx-spring-boot-starter
- dev.snowdrop.vertx-spring-boot-starter-actuator
- dev.snowdrop.vertx-spring-boot-starter-http
- dev.snowdrop.vertx-spring-boot-starter-http-test
- dev.snowdrop.vertx-spring-boot-starter-mail
- io.dekorate

4.4. ARTIFACTS REMOVED IN THIS RELEASE

No supported artifacts are removed in this release.

4.5. ARTIFACTS DEPRECATED IN THIS RELEASE

No artifacts are marked as deprecated in this release.
CHAPTER 5. FIXED ISSUES

This RHOAR Spring Boot release incorporates all bugfixes from the upstream release. Issues resolved in the community release are listed in the Spring Boot 2.1.6 Release Notes.
6.1. **SB-379**: MISSING APR/NATIVE LIBRARY IN THE `openshift-openjdk` IMAGE

6.2. CONNECTION BETWEEN A RHEL 8-BASED DATABASE APPLICATION AND A RHEL 7-BASED MYSQL 5.7 DATABASE FAILS DUE TO TLS PROTOCOL VERSION MISMATCH

**Description**

Attempting to open a TLS-secured connection using OpenSSL between a database application container built on a RHEL 8-based OpenJDK builder image a database container built on a RHEL 7-based MySQL 5.7 container image results in a connection failure due to a `javax.net.ssl.SSLHandshakeException` at runtime: For more detail, view the issue in JIRA.

```java
Caused by: javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)
```

**Cause**

The issue occurs due to a difference in the latest supported TLS protocol version between RHEL 7 and RHEL 8. The TLS implementation on RHEL 7 supports TLS protocol versions 1.0 (deprecated), 1.1, and 1.2. The TLS implementation on RHEL 8 also supports TLS protocol version 1.3, which is also the default TLS version used in RHEL 8-based builder images. This discrepancy may cause a TLS protocol version mismatch between application components while negotiating a TLS handshake, which in turn causes the connection between the application and database containers to fail.

**Workaround**

To prevent the issue described above, manually specify a TLS protocol version that is supported on both operating system versions in your database connection string. For example:

```sql
jdbc:mysql://testdb-mysql:3306/testdb?enabledTLSProtocols=TLSv1.2
```
CHAPTER 7. KNOWN ISSUES AFFECTING REQUIRED INFRASTRUCTURE COMPONENTS

- **Fabric8 Maven Plugin Issue #1640**: Pushing an image into a custom repository during an s2i build with FMP 4.1.0 results in a `DuplicateKeyException`.

**Affected components and component versions**

This issue affects Fabric8 Maven Plugin 4.1.0.

**Description**

Fabric8 Maven Plugin does not process `ImageConfiguration` unless `ImageConfiguration` also contains a `BuildImageConfiguration`. Without a recognizable `BuildImageConfiguration`, Fabric8 Maven Plugin repeatedly calls the s2i image generators to create another default `ImageConfiguration` that contains the expected `BuildImageConfiguration`. This results in more than one `ImageConfiguration` being specified for the given s2i build, which in turn results in a `DuplicateKeyException` when FMP attempts to push the image to the registry specified in the `pom.xml` configuration file. This leads to image build failures when a new image build is triggered by a change in the deployment configuration of a pod on OpenShift.

**Workaround**

To prevent Fabric8 Maven Plugin from generating a duplicate `ImageConfiguration`, place the image configuration inside a `build` section in the `pom.xml` configuration file of your project, as shown in the examples below. This in turn prevents the `DuplicateKeyException` when a new image build is triggered by a change in the deployment configuration of the pod.

```xml
<configuration>
  <images>
    <image>
      <name>
        artifactrepository.somecompany.com:18444/demo-boot/demo-boot:1.0
      </name>
      <build>
        <from>
          fabric8/S2I_BASE_IMAGE_NAME
        </from>
        <assembly>
          <basedir>/deployments
          </basedir>
          <descriptorRef>artifact-with-dependencies</descriptorRef>
        </assembly>
        <env>
          <JAVA_LIB_DIR>
          /deployments
          </JAVA_LIB_DIR>
          <JAVA_MAIN_CLASS>org.example.class.name.Main</JAVA_MAIN_CLASS>
        </env>
      </build>
    </image>
  </images>
</configuration>
```
CHAPTER 7. KNOWN ISSUES AFFECTING REQUIRED INFRASTRUCTURE COMPONENTS

</images>
...
</configuration>
CHAPTER 8. ADVISORIES RELATED TO THIS RELEASE

The following advisories have been issued to document enhancements, bugfixes, and CVE fixes included in this release.

- RHBA-2019:2653