Red Hat OpenShift Application Runtimes 1

RHOAR Eclipse Vert.x Release Notes

For use with Red Hat OpenShift Application Runtimes Eclipse Vert.x 3.7.1

Last Updated: 2019-07-08

Red Hat OpenShift Application Runtimes 1 RHOAR Eclipse Vert.x Release Notes

For use with Red Hat OpenShift Application Runtimes Eclipse Vert.x 3.7.1

Legal Notice

Abstract

This Release Note contains important information related to Red Hat OpenShift Application Runtimes Eclipse Vert.x

Table of Contents

PREFACE

Date of release: 2019-03-14

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.

Component name	Version
Maven	3.5.0
Fabric8 Maven Plugin	4.1.0
JDK[a][b]	OpenJDK 8, OpenJDK 11[c]
OpenShift Container Platform (OCP)[d]	3.11 or later
Minishift	1.34.0 or later
CDK[e]	3.8.0
git	2.0 or later
oc command line tool	3.11 or later[f]

[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] OCP is supported by Red Hat

[e] CDK is supported by Red Hat

[f] The version of the **oc** CLI tool should correspond to the version of OCP that you are using.

CHAPTER 2. SUPPORTED ECLIPSE VERT.X RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

The following resource defines the supported configurations and integrations of Red Hat products with RHOAR Eclipse Vert.x:

- For a list of technologies that are supported for integration with RHOAR Eclipse Vert.x in production environments see the Supported Eclipse Vert.x configurations and integrations.
- For a list of RHOAR Eclipse Vert.x runtime artifacts and their versions see the Eclipse Vert.x 3.7.1 component details page.

CHAPTER 3. FEATURES

3.1. NEW AND CHANGED FEATURES

This release of RHOAR Eclipse Vert.x introduces the following new features and feature updates:

- The **micrometer-registry-prometheus** artifact is set as the default in the **vertx-micrometer-metrics** artifact. If you use Prometheus to expose application metrics, you no longer need to include the **micrometer-registry-prometheus** artifact as a dependency in the **pom.xml** file of your application.
- Red Hat supports Agroal as the default database connection pool with the **vertx-jdbc-client** artifact. Agroal was set as the default connection pool in the Vert.x 3.5.1 release.
- Infinispan is upgraded to 9.4.6.Final-redhat-00002.

3.2. DEPRECATED FEATURES

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

CHAPTER 4. RELEASE COMPONENTS

4.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE

AMQ client

This release supports sending and receiving AMQP messages through the **vertx-amqp-client** artifact. The **vertx-amqp-client** replaces the AMQP bridge and provides a more user-friendly API.

Kafka client

This release supports the **vertx-kafka-client** artifact for sending and receiving messages to Apache Kafka clusters.

4.2. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE

The following artifacts are provided as Technology Preview in this release.

Vert.x Config Vault

This release includes the **vertx-config-vault** artifact that extends the Vert.x Configuration Retriever to support retrieving configuration secrets from Vault.

Vert.x Redis Client

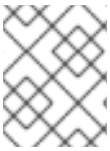
This release includes the **vertx-redis-client** artifact which is revised to provide an updated API.

Vert.x Web API Contract

This release includes the **vertx-web-api-contract** artifact that extends Vert.x Web to support OpenAPI 3.

Vert.x Web GraphQL

This release includes the **vertx-web-graphql** artifact that extends Vert.x Web capabilities and enables you to build a GraphQL server.



NOTE

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

4.3. ARTIFACTS REMOVED IN THIS RELEASE

No artifacts are removed in this release.

4.4. ARTIFACTS DEPRECATED IN THIS RELEASE

No artifacts are marked as deprecated in this release.

CHAPTER 5. FIXED ISSUES

This RHOAR Eclipse Vert.x release incorporates all bugfixes from the upstream release. Issues resolved in the community release are listed in the Eclipse Vert.x 3.7.1 Release Notes .

CHAPTER 6. KNOWN ISSUES

6.1. MISSING STARR.VERSION WARNING IN PROJECTS THAT INCLUDE THE VERTX-KAFKA-CLIENT ARTIFACT

When building a project that includes the **vertx-kafka-client** artifact, Maven generates a warning.

```
The POM for org.scala-lang:scala-compiler:jar:${starr.version} is missing, no dependency information available
```

To avoid this warning and a potential build failure, add the **-Dstarr.version** property to your build command.

```
-Dstarr.version=2.12.8.redhat-00001
```

6.2. FALSE CONNECTION RESET BY PEER ERROR MESSAGES WHEN CALLING APPLICATION ENDPOINT

Making an HTTP request on an endpoint of a Vert.x application using either curl or a Java HTTP client, produces the following error in the logs after each request:

```
io.vertx.core.net.impl.ConnectionBase
SEVERE: java.io.IOException: Connection reset by peer
```

This behavior is caused by the interaction of the Netty application framework and the HAProxy load-balancer used by OpenShift. The error occurs due to existing HTTP connections being re-used by HAProxy without closing. Even though the error message is logged, no error condition occurs. HTTP requests are handled correctly and the application responds as expected.

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 issues 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 form 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 new image build is triggered by a change in the deployment configuration of the pod.

```
<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>
  ...
</configurtation>
```

CHAPTER 8. ADVISORIES RELATED TO THIS RELEASE

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

- RHEA-2019:1473