Red Hat build of Eclipse Vert.x 4.0

Release Notes for Eclipse Vert.x 4.0

For use with Eclipse Vert.x 4.0.3

Last Updated: 2021-03-29

Red Hat build of Eclipse Vert.x 4.0 Release Notes for Eclipse Vert.x 4.0

For use with Eclipse Vert.x 4.0.3

Legal Notice

Abstract

This Release Note contains important information related to Eclipse Vert.x 4.0.3

Table of Contents

PREFACE

Date of release: 2021-03-29

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments.

This section explains how to submit feedback.

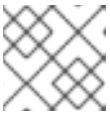
Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in **Multi-page HTML** format.

Procedure

To provide your feedback, perform the following steps:

1. Click the **Feedback** button in the top-right corner of the document to see existing feedback.



NOTE

The feedback feature is enabled only in the **Multi-page HTML** format.

2. Highlight the section of the document where you want to provide feedback.
3. Click the **Add Feedback** pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.
4. Enter your feedback in the text box and click **Submit**.
   A documentation issue is created.
5. To view the issue, click the issue tracker link in the feedback view.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. REQUIRED INFRASTRUCTURE COMPONENT VERSIONS

Red Hat does not provide support for components listed below, with the exception of components explicitly designated as supported.

Component name	Version
Maven	3.6.0
OpenShift Maven Plugin	1.1.1
JDK[a][b]	OpenJDK 8 or OpenJDK 11[c]
Red Hat Enterprise Linux 7[d]	7.9
Red Hat Enterprise Linux 8[e]	8.3
OpenShift Container Platform (OCP)[f]	3.11, 4.7
Minishift	1.34.3 or later
CDK[g]	3.11.0
git	2.0 or later
oc command line tool	3.11 or later[h]

[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] Red Hat supports only LTS releases of JDK.

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

[e] For deploying applications based on CNR 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 ECLIPSE VERT.X RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

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

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

CHAPTER 3. FEATURES

3.1. NEW AND CHANGED FEATURES

This section describes the new functionalities introduced in this release. It also contains information about changes in the existing functionalities.

3.1.1. Context server interceptor binds all types of data and is more secure

From Eclipse Vert.x 4.0.3, the **ContextServerInterceptor.bind()** method binds all types of data to the context. The method is more secure now as it does not expose the storage details.

In releases prior to Eclipse Vert.x 4.0.3, the method used to bind only 'String' data type to context. It also exposed the storage details.

To use the updated **ContextServerInterceptor.bind()** method, you must update your application.

The following example shows the code in releases prior to Eclipse Vert.x 4.0.3.

```
// Example code from previous releases

class X extends ContextServerInterceptor {
  @Override
  public void bind(Metadata metadata, ConcurrentMap<String, String> context) {
```

The following example shows the replacing code fpr Eclipse Vert.x 4.0.3 release.

```
// Replacing code for Eclipse Vert.x 4.0.3 release

class X extends ContextServerInterceptor {
  @Override
  public void bind(Metadata metadata) {
```

3.1.2. Matching of ending slash (/) in route paths that end with wildcard character is no longer required

In releases prior to Eclipse Vert.x 4.0.3, if routes were defined with a path ending in slash and a wildcard /*, the routes would be called only if the matching request also included the ending slash /. This rule caused problems when the wildcard was empty.

From Eclipse Vert.x 4.0.3 onward, this rule is no longer applied. You can create routes whose paths end in a slash (/). However, it is not mandatory to specify the slash in the request URLs.

Also, you can create and use request URLs to call routes that end with wildcards in their path instead of slash (/). For example, routes with wildcard can be defined as **/foo/***. Here the route has to match an open wildcard at the end of the path. The request URL can be **/foo**.

The table shows the behavior in Eclipse Vert.x 4.0.3 and previous releases when you send a request URL **/foo/***. You can see that the ending slash is optional in Eclipse Vert.x 4.0.3 and request matches the route.

Route	Eclipse Vert.x 4.0.3	Releases prior to Eclipse Vert.x 4.0.3
/foo	Match	No Match
/foofighters	No Match	No Match
/foo/	Match	Match
/foo/bar	Match	Match

3.1.3. Removed the autoRegistrationOfImporters attribute from service discovery options

The **autoRegistrationOfImporters** attribute has been removed from service discovery options.

3.1.4. Authenticate method in authentication provider class updated to support token as input credentials

In releases prior to Eclipse Vert.x 4.0.3, the AuthenticationProvider.authenticate() method would incorrectly take **jwt: someValue** as input credentials.

From Eclipse Vert.x 4.0.3, the AuthenticationProvider.authenticate() method has been updated and takes **token: someValue** as input credentials. This change ensures that both JSON and typed APIs are consistent and can be used interchangeably.

The following code shows the implementation for the authenticate method in releases prior to Eclipse Vert.x 4.0.3.

```
new JsonObject().put("jwt", "token...");
```

The following code shows the implementation for the authenticate method in Eclipse Vert.x 4.0.3 release.

```
new JsonObject().put("token", "token...");
```

3.1.5. Get method for PEM keys returns Buffer instead of a String

The **PubSecKeyOptions.getBuffer()** method returns the PEM or secret key buffer. In releases prior to Eclipse Vert.x 4.0.2, the key buffer was stored and returned as a **String**. However, it is recommended to save secrets as a **Buffer**. From Eclipse Vert.x 4.0.2 onward, the method stores and returns the key buffer as a **Buffer**. This change improves the security and handling of secrets.

The **PubSecKeyOptions.setBuffer()** method continues to accept a **String** argument. In the set method, an overload for Buffer has been added to safely handle non ASCII secret materials. This change does not require any change to the existing code.

3.1.6. Kubernetes service importer is no longer registered automatically

From Eclipse Vert.x 4, the **KubernetesServiceImporter** discovery bridge is no longer registered automatically. Even if you have added the bridge in the classpath of your Maven project, it will not be automatically registered.

You must manually register the bridge after creating the **ServiceDiscovery** instance.

3.1.7. Use future methods for asynchronous operations

Eclipse Vert.x 4 uses futures for asynchronous operations. Every callback method has a corresponding future method.

Futures can be used to compose asynchronous operations. When you use futures, the error handling is better. Therefore, it is recommended to use a combination of callback and futures in your applications.

3.1.8. No dependency on the Jackson Databind library

In Eclipse Vert.x 4, Jackson Databind is an optional Maven dependency. If you want to use this dependency, you must explicitly add it in the classpath. For example, if you are object mapping JSON, then you must explicitly add the dependency.

3.1.9. Handling deprecations and removals

In Eclipse Vert.x 4, new enhanced features have been provided. The old features and functions have been deprecated or removed in Eclipse Vert.x 4. Before you migrate your applications to Eclipse Vert.x 4, check for deprecations and removals.

The Java compiler generates warnings when deprecated APIs are used. You can use the compiler to check for deprecated methods while migrating applications to Eclipse Vert.x 4.

3.1.10. Support for distributed tracing

Eclipse Vert.x 4 supports distributed tracing. You can use tracing to monitor microservices and identify performance issues.

Eclipse Vert.x 4 integrates with OpenTracing system.

The following Eclipse Vert.x components can log traces:

- HTTP server and HTTP client
- Eclipse Vert.x SQL client
- Eclipse Vert.x Kafka client

IMPORTANT

Tracing is available as Technology Preview. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend to use them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

See Technology Preview Features Support Scope on the Red Hat Customer Portal for information about the support scope for Technology Preview features.

3.1.11. New publishing location for EventBus JavaScript Client

In Eclipse Vert.x 4, the EventBus JavaScript client, **vertx-web-client.js** is not published as a Red Hat artifact in the Maven repository.

The client is published in the npm repository. You can access the client from the following location: @vertx/eventbus-bridge-client.js

3.1.12. Deploy Eclipse Vert.x applications using OpenShift Maven plugin

Use the OpenShift Maven plugin to deploy your Eclipse Vert.x applications on OpenShift. The Fabric8 Maven plugin is no longer supported. For more information, see the section migrating from Fabric8 Maven Plugin to Eclipse JKube.

3.1.13. Eclipse Vert.x metering labels for OpenShift

You can add metering labels to your Eclipse Vert.x pods and check Red Hat subscription details with the OpenShift Metering Operator.

NOTE

Do not add metering labels to any pods that an operator deploys and manages.

Eclipse Vert.x should use the following metering labels:

- **com.redhat.component-name: Vert.x**
- **com.redhat.component-type: application**
- **com.redhat.component-version: 4.0.3**
- **com.redhat.product-name: "Red_Hat_Runtimes"**
- **com.redhat.product-version: 2021-Q1**

See Metering documentation for more information.

For more information on labels, see Understanding how to update labels on nodes .

3.1.14. Support for OpenJDK 8 and OpenJDK 11 RHEL 8 Universal Base Images (UBI8)

Eclipse Vert.x introduces support for building and deploying Eclipse Vert.x applications to OpenShift with OCI-compliant Universal Base Images for Red Hat OpenJDK 8 and Red Hat OpenJDK 11 on RHEL 8.

The RHEL 8 OpenJDK Universal Base Images replace the RHEL 8 OpenJDK builder images. The RHEL 8 OpenJDK base images are no longer supported for use with Eclipse Vert.x.

3.2. DEPRECATED FEATURES

This section lists the functionalities deprecated or removed in this release.

- **HttpServerOptions**

Removed methods	Replacing methods
getMaxWebsocketFrameSize()	**getMaxWebSocketFrameSize()**
setMaxWebsocketFrameSize()	**setMaxWebSocketFrameSize()**
getMaxWebsocketMessageSize()	**getMaxWebSocketMessageSize()**
setMaxWebsocketMessageSize()	**setMaxWebSocketMessageSize()**
getPerFrameWebsocketCompressionSupported()	**getPerFrameWebSocketCompressionSupported()**
setPerFrameWebsocketCompressionSupported()	**setPerFrameWebSocketCompressionSupported()**
getPerMessageWebsocketCompressionSupported()	**getPerMessageWebSocketCompressionSupported()**
setPerMessageWebsocketCompressionSupported()	**setPerMessageWebSocketCompressionSupported()**
getWebsocketAllowServerNoContext()	**getWebSocketAllowServerNoContext()**
setWebsocketAllowServerNoContext()	**setWebSocketAllowServerNoContext()**
getWebsocketCompressionLevel()	**getWebSocketCompressionLevel()**
setWebsocketCompressionLevel()	**setWebSocketCompressionLevel()**
getWebsocketPreferredClientNoContext()	**getWebSocketPreferredClientNoContext()**
setWebsocketPreferredClientNoContext()	**setWebSocketPreferredClientNoContext()**
getWebsocketSubProtocols()	**getWebSocketSubProtocols()**
setWebsocketSubProtocols()	**setWebSocketSubProtocols()**

- Eclipse Vert.x Web

Removed elements	Replacing elements
io.vertx.ext.web.Cookie	**io.vertx.core.http.Cookie**
io.vertx.ext.web.handler.CookieHandler	**io.vertx.core.http.Cookie**

Removed elements	Replacing elements
io.vertx.ext.web.Locale	io.vertx.ext.web.LanguageHeader
RoutingContext.acceptableLocales()	RoutingContext.acceptableLanguages()
StaticHandler.create(String, ClassLoader)	---
SessionHandler.setAuthProvider(AuthProvider)	SessionHandler.addAuthProvider()
HandlebarsTemplateEngine.getHandlebars()HandlebarsTemplateEngine.getResolvers()HandlebarsTemplateEngine.setResolvers()JadeTemplateEngine.getJadeConfiguration()ThymeleafTemplateEngine.getThymeleafTemplateEngine()ThymeleafTemplateEngine.setMode()	TemplateEngine.unwrap()

- Messaging

Removed methods	Replacing methods
MessageProducer<T>.send(T)	MessageProducer<T>.write(T)
MessageProducer.send(T,Handler)	EventBus.request(String,Object,Handler)

- EventBus

Removed methods	Replacing methods
EventBus.send(…, Handler<AsyncResult<Message<T>>>)Message.reply(…, Handler<AsyncResult<Message<T>>>)	replyAndRequest

- Handlers

Removed methods	Replacing methods
Future<T>.setHandler()	Future<T>.onComplete()Future<T>.onSuccess()Future<T>.onFailure()
HttpClientRequest.connectionHandler()	HttpClient.connectionHandler()

- JSON

Removed Fields/Methods	New methods
Json.mapper() field	**DatabindCodec.mapper()**
Json.prettyMapper() field	**DatabindCodec.prettyMapper()**
Json.decodeValue(Buffer, TypeReference<T>)	**JacksonCodec.decodeValue(Buffer, TypeReference)**
Json.decodeValue(String, TypeReference<T>)	**JacksonCodec.decodeValue(String, TypeReference)**

- JUnit5

Deprecated methods	New methods
VertxTestContext.succeeding()	**VertxTestContext.succeedingThenComplete()**
VertxTestContext.failing()	**VertxTestContext.failingThenComplete()**

- Reactive Extensions (Rx)

Deprecated methods	New methods
WriteStreamSubscriber.onComplete()	**WriteStreamSubscriber.onWriteStreamEnd()WriteStreamSubscriber.onWriteStreamError()**

- Circuit breaker

Removed methods	Replacing methods
CircuitBreaker.executeCommand()	**CircuitBreaker.execute()**
CircuitBreaker.executeCommandWithFallback()	**CircuitBreaker.executeWithFallback()**

- MQTT

Removed methods	Replacing methods
MqttWill.willMessage()	**MqttWill.getWillMessage()**
MqttWill.willTopic()	**MqttWill.getWillTopic()**
MqttWill.willQos()	**MqttWill.getWillQos()**

Removed methods	Replacing methods
MqttAuth.username()	**MqttAuth.getUsername()**
MqttAuth.password()	**MqttAuth.getPassword()**
MqttClientOptions.setKeepAliveTimeSeconds()	**MqttClientOptions.setKeepAliveInterval()**

- AMQP client

Removed methods	Replacing methods
AmqpClient.createReceiver(String address, Handler<AmqpMessage> messageHandler, …)	**AmqpClient createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)**
AmqpConnection createReceiver(…, Handler<AsyncResult<AmqpReceiver>> completionHandler)	**AmqpConnection createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)**
AmqpConnection createReceiver(.., Handler<AmqpMessage> messageHandler, Handler<AsyncResult<AmqpReceiver>> completionHandler)	**AmqpConnection createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)**

- Authentication and authorization

Removed elements	Replacing elements
OAuth2Options.isUseBasicAuthorizationHeader()	No replacing method
OAuth2Options.setUseBasicAuthorizationHeader()	No replacing method
OAuth2Options.getClientSecretParameterName()	No replacing method
OAuth2Options.setClientSecretParameterName()	No replacing method
OAuth2Auth.createKeycloak()	**KeycloakAuth.create(vertx, JsonObject)()**

Removed elements	Replacing elements
OAuth2Auth.create(Vertx, OAuth2FlowType, OAuth2ClientOptions) ()	**OAuth2Auth.create(vertx, new OAuth2ClientOptions().setFlow(YOUR_DESIRED_FLOW))**
OAuth2Auth.create(Vertx, OAuth2FlowType)	**OAuth2Auth.create(vertx, new OAuth2ClientOptions().setFlow(YOUR_DESIRED_FLOW))**
User.isAuthorised()	**User.isAuthorized()**
AccessToken.refreshToken()	**AccessToken.opaqueRefreshToken()**
io.vertx.ext.auth.jwt.JWTOptions data object	**io.vertx.ext.jwt.JWTOptions** data object
SecretOptions class	**PubSecKeyOptions** class

Deprecated methods	Replacing methods
OAuth2Auth.decodeToken()	**AuthProvider.authenticate()**
OAuth2Auth.introspectToken()	**AuthProvider.authenticate()**
OAuth2Auth.getFlowType()	No replacing method
OAuth2Auth.loadJWK()	**OAuth2Auth.jwkSet()**
Oauth2ClientOptions.isUseAuthorizationHeader()	No replacing method

Deprecated class	Replacing class
AbstractUser	Create user objects using the ` User.create(JsonObject)` method.
AuthOptions	No replacing class
JDBCAuthOptions	**JDBCAuthenticationOptions** for authentication and **JDBCAuthorizationOptions** for authorization

Deprecated class	Replacing class
JDBCHashStrategy	No replacing class
OAuth2RBAC	**AuthorizationProvider**
Oauth2Response	Recommended to use **WebClient** class
KeycloakHelper	No replacing class

- Service discovery

Removed methods	Replacing methods
ServiceDiscovery.create(…, Handler<ServiceDiscovery> completionHandler)	**ServiceDiscovery.create(Vertx)**
ServiceDiscovery.create(…, Handler<ServiceDiscovery> completionHandler)	**ServiceDiscovery.create(Vertx, ServiceDiscoveryOptions)**

- Eclipse Vert.x configuration

Removed methods	Replacing methods
ConfigRetriever.getConfigAsFuture()	**retriever.getConfig()**

- MongoDB client

Removed methods	Replacing methods
MongoClient.update()	**MongoClient.updateCollection()**
MongoClient.updateWithOptions()	**MongoClient.updateCollectionWithOptions()**
MongoClient.replace()	**MongoClient.replaceDocuments()**
MongoClient.replaceWithOptions()	**MongoClient.replaceDocumentsWithOptions()**
MongoClient.remove()	**MongoClient.removeDocuments()**
MongoClient.removeWithOptions()	**MongoClient.removeDocumentsWithOptions()**
MongoClient.removeOne()	**MongoClient.removeDocument()**

Removed methods	Replacing methods
MongoClient.removeOneWithOptions	**MongoClient.removeDocumentsWithOptions()**

- Clients with no shared data sources

Deprecated Methods	New Methods
MongoClient.createNonShared()	**MongoClient.create()**
JDBCClient.createNonShared()	**wJDBCClient.create()**
CassandraClient.createNonShared()	**CassandraClient.create()**
MailClient.createNonShared()	**MailClient.create()**

- Hook methods

Removed Methods	New Methods
Context.addCloseHook()	No replacing method
Context.removeCloseHook()	No replacing method

- Clone methods

Removed Methods	New Methods
KeyCertOptions.clone()	**KeyCertOptions.copy()**
TrustOptions.clone()	**TrustOptions.copy()**
SSLEngineOptions.clone()	**SSLEngineOptions.copy()**

- **VertxOptions**

Removed Methods	New Methods
VertxOptions.equals()	No replacing method
VertxOptions.hashCode()	No replacing method
VertxOptions.fileResolverCachingEnabled()	**FileSystemOptions.isFileCachingEnabled()**

- Pooled buffer

Removed Methods	New Methods
TCPSSLOptions.isUsePooledBuffers()	No replacing method
TCPSSLOptions.setUsePooledBuffers()	No replacing method

CHAPTER 4. RELEASE COMPONENTS

4.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE

No artifacts have been moved from Technology Preview to fully supported in this release:

4.2. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE

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

- **vertx-auth-webauthn**
  The Eclipse Vert.x authentication module **io.vertx.ext.auth.AuthProvider** interface has been split into two new interfaces:
  - **io.vertx.ext.auth.authentication.AuthenticationProvider**
  - **io.vertx.ext.auth.authorization.AuthorizationProvider**

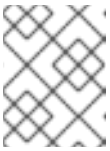
Authentication is a new feature in Eclipse Vert.x 4. In earlier releases, you could only check if a user was authorized to perform the tasks on the **User** object. This meant that the provider was responsible for both authentication and authorization of the user.

In Eclipse Vert.x 4, the **User** object instances are not associated with a particular authentication provider. So you can authenticate and authorize a user using different providers.

- **vertx-opentracing**
  Eclipse Vert.x 4 supports distributed tracing. You can use tracing to monitor microservices and identify performance issues.

  Eclipse Vert.x 4 integrates with OpenTracing system.

  The following Eclipse Vert.x components can log traces:
  - HTTP server and HTTP client
  - Eclipse Vert.x SQL client
  - Eclipse Vert.x Kafka client



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 Eclipse Vert.x release incorporates all bugfixes from community release of version 4.0.3. Issues resolved in the community release are listed in the Eclipse Vert.x 4.0.3.

5.1. GOOGLE GUAVA CLASSES INCLUDED IN GRAPHQL BUILDS

In the Eclipse Vert.x 4.0.0 and 4.0.2 releases, the **vertx-web-graphql** dependency was not usable. This was because an incomplete build of GraphQL Java with version 16.1.0.redhat-00001 was used. In the incomplete GraphQL build, the Guava classes were missing.

This issue is resolved in the Eclipse Vert.x 4.0.3 release. The release includes the GraphQL Java 16.1.0.redhat-00002 version, which is a complete build with Guava classes. These Guava classes are shaded into the jar.

5.2. VERTX-OPENTRACING AVAILABLE IN ECLIPSE VERT.X BUILDS

The **vertx-opentracing** dependency was introduced as a Technical Preview feature in Eclipse Vert.x 4.0.0. However, the dependency was not available in Eclipse Vert.x 4.0.0 and 4.0.2 releases.

This issue is resolved in Eclipse Vert.x 4.0.3 release. The release includes the **vertx-opentracing** dependency.

CHAPTER 6. KNOWN ISSUES

6.1. KUBERNETESSERVICEIMPORTER() CANNOT BE DIRECTLY REGISTERED IN ECLIPSE VERT.X REACTIVE EXTENSIONS (RX)

Description

You cannot directly register **KubernetesServiceImporter()** with the Reactive Extensions (Rx) for Eclipse Vert.x.

Cause

Service importers do not have a generated RxJava 2 implementation.

Workaround

You must create an instance of **KubernetesServiceImporter** and encapsulate it with **{@link io.vertx.reactivex.servicediscovery.spi.ServiceImporter}** as shown in the following example:

```
{@link
examples.RxServiceDiscoveryExamples#register(io.vertx.reactivex.servicediscovery.ServiceDiscovery)}
```

The following example shows how to register **KubernetesServiceImporter()** in Eclipse Vert.x Reactive Extensions (Rx).

```
ServiceDiscovery discovery = ServiceDiscovery.create(vertx);
discovery.getDelegate().registerServiceImporter(new KubernetesServiceImporter(), new
JsonObject());
```

6.2. RED HAT AMQ STREAMS IMAGES ARE NOT AVAILABLE FOR IBM Z AND IBM POWER SYSTEMS

The Red Hat AMQ Streams Operator and Kafka images are not available for IBM Z and IBM Power Systems. Since the images are not available, the **vertx-kafka-client** module is not certified to work with AMQ Streams on IBM Z and IBM Power Systems.

6.3. 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 an application container built on a RHEL 8-based OpenJDK builder image and 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 .

```
...
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:

```
jdbc:mysql://testdb-mysql:3306/testdb?enabledTLSProtocols=TLSv1.2
```

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

Making an HTTP request on an endpoint of an Eclipse Vert.x application using either the **curl** tool 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. ADVISORIES RELATED TO THIS RELEASE

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

- RHSA-2021:0943