



Red Hat build of Quarkus 1.7

Release Notes for Red Hat build of Quarkus 1.7

Red Hat build of Quarkus 1.7 Release Notes for Red Hat build of Quarkus 1.7

Legal Notice

Copyright © 2021 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.

Abstract

This document contains release notes for Red Hat build of Quarkus 1.7.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. RED HAT BUILD OF QUARKUS	5
CHAPTER 2. NEW AND CHANGED FEATURES	6
2.1. GRPC QUARKUS EXTENSION	6
2.2. QUARKUS NATIVE MODE	6
2.3. OPENSIFT SERVERLESS SERVING IN NATIVE AND JVM MODE	6
2.4. SUPPORT FOR CLIENT TO CONNECT TO RED HAT DATA GRID	6
2.5. SPRING COMPATIBILITY LAYERS	7
CHAPTER 3. RED HAT BUILD OF QUARKUS SUPPORTED PLATFORMS, CONFIGURATIONS, EXTENSIONS, AND DEPENDENCIES	8
3.1. SUPPORTED EXTENSIONS AND DEPENDENCIES	8
3.2. DEVELOPMENT SUPPORT	8
CHAPTER 4. DEPRECATED COMPONENTS AND FEATURES	9
CHAPTER 5. TECHNOLOGY PREVIEW	10
CHAPTER 6. KNOWN ISSUES	11
CHAPTER 7. FIXED ISSUES IN QUARKUS 1.7.5 SP1	12
CHAPTER 8. FIXED ISSUES IN QUARKUS 1.7.6 GA	13
8.1. SECURITY-RELATED BUG FIXES	13
8.2. CHANGES RELATED TO INTEGRATION WITH OTHER PRODUCTS AND TECHNOLOGIES	13
8.3. DEVELOPER-EXPERIENCE-RELATED BUG FIXES	13
8.4. COMPONENT VERSION UPGRADES	14

PREFACE

These release notes list new features, features in technology preview, known issues, and issues fixed in Red Hat build of Quarkus 1.7.

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. RED HAT BUILD OF QUARKUS

Red Hat build of Quarkus is a Kubernetes-native Java stack that is optimized for use with containers and Red Hat OpenShift Container Platform. Quarkus is designed to work with popular Java standards, frameworks, and libraries such as Eclipse MicroProfile, Apache Kafka, RESTEasy (JAX-RS), Hibernate ORM (JPA), Spring, Infinispan, and Apache Camel.

The Quarkus dependency injection solution is based on CDI (contexts and dependency injection) and includes an extension framework to expand functionality and to configure, boot, and integrate a framework into your application.

Quarkus provides a container-first approach to building Java applications. This approach makes it much easier to build microservices-based applications written in Java as well as enabling those applications to invoke functions running on serverless computing frameworks. For this reason, Quarkus applications have small memory footprints and fast startup times.

CHAPTER 2. NEW AND CHANGED FEATURES

This section provides an overview of new features and changes introduced in Red Hat build of Quarkus 1.7.6.

2.1. GRPC QUARKUS EXTENSION

The **quarkus-grpc** extension introduces the following capabilities:

- Ability for Quarkus application to consume and expose gRPC services.
- Support based on the Quarkus reactive core as well as imperative (blocking) consumption.
- Support for plain-text and TLS communication, both relying on HTTP/2 transport.
- Support for TLS encryption and mutual authentication.
- Implementation of the gRPC health and reflection services. Health is integrated with the **quarkus-smallrye-health** extension and the reflection service makes it easier to use tools such as `grpcurl`.

2.2. QUARKUS NATIVE MODE

This release provides full support for running a Quarkus application as a native application when compiled using the Red Hat build of Quarkus Native build container. Running a Quarkus application as a native application instead of a Java application can greatly improve startup times and memory utilization. This feature was Tech preview in the previous version of Red Hat build of Quarkus.

2.3. OPENSIFT SERVERLESS SERVING IN NATIVE AND JVM MODE

Running a Quarkus application in OpenShift Serverless (Serving) is now fully supported in both JVM and native mode. The **quarkus-openshift** extension additionally supports deploying a JVM applications to OpenShift Container Platforms with Serverless Serving installed using the Apache Maven.

2.4. SUPPORT FOR CLIENT TO CONNECT TO RED HAT DATA GRID

With the **quarkus-infinispan-client** extension the client can connect to a server running outside of application processes.

The Infinispan client Quarkus extension provides the following features:

- Creation of native image
- Automatic injection of important resources such as RemoteCache (named), RemoteCacheManger, CounterManager
- User based ProtoStream Marshalling
- Indexed and non-indexed querying
- Continuous query
- Near cache

- Authentication and authorization
- Encryption
- Counters

2.5. SPRING COMPATIBILITY LAYERS

Expanded support for new Spring compatibility layers including:

- **quarkus-spring-cache**
- **quarkus-spring-cloud-config-client**
- **quarkus-spring-scheduled**

CHAPTER 3. RED HAT BUILD OF QUARKUS SUPPORTED PLATFORMS, CONFIGURATIONS, EXTENSIONS, AND DEPENDENCIES

- For a list of supported configurations and tested integrations see the [Red Hat build of Quarkus Supported Configurations](#) page (login required).
- For a list of supported Maven artifacts see the [Red Hat build of Quarkus Component Details](#) page (login required).

3.1. SUPPORTED EXTENSIONS AND DEPENDENCIES

For a list of Red Hat build of Quarkus extensions and dependencies that Red Hat supports for use in production environments see the [Red Hat build of Quarkus Component Details](#) page (login required).

3.2. DEVELOPMENT SUPPORT

Red Hat provides [development support](#) for the following Red Hat build of Quarkus features, plug-ins, extensions, and dependencies:

Features

- Live development mode
- Remote development mode

Plug-ins

- **protobuf-maven-plugin**

Extensions and dependencies

- For a list of Red Hat build of Quarkus extensions that Red Hat supports under the [development support](#) scope of coverage see the [Red Hat build of Quarkus Component Details](#) page (login required).

CHAPTER 4. DEPRECATED COMPONENTS AND FEATURES

The components and features listed in this section are deprecated with Red Hat build of Quarkus 1.7. They are included and supported in this release, however no enhancements will be made to these components and features and they might be removed in the future.

- The **quarkus-smallrye-opentracing** extension
- The use of ReactiveX APIs

CHAPTER 5. TECHNOLOGY PREVIEW

This section lists features and extensions that are in Technology Preview in Red Hat build of Quarkus 1.7.



IMPORTANT

These features are for Technology Preview only. 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.

For more information on Red Hat Technology Preview features, see [Technology Preview Features Scope](#).

Extensions and dependencies

- For a list of Red Hat build of Quarkus extensions and dependencies that Red Hat provides as Technology Preview with Red Hat build of Quarkus 1.7.6 see the [Red Hat build of Quarkus Component Details](#) page (login required).

CHAPTER 6. KNOWN ISSUES

This section lists known issues with Red Hat build of Quarkus 1.7.

- [Issue #11633](#) Missing zero-config solution for OpenShift Serverless. This issue affects only deployment of Quarkus native Serverless applications
- [QUARKUS-451](#) Increased instability in **NativeAggregatorIT** in **kafka-streams-quickstart**
- [QUARKUS-688](#) Cannot deploy **mutable-jar** (used by remote development feature) on OpenShift

CHAPTER 7. FIXED ISSUES IN QUARKUS 1.7.5 SP1

Quarkus 1.7.5 SP1 contains the following bug fix:

- For Quarkus 1.7.5 SP1, a vulnerability was fixed in Hibernate ORM, where the JPA Criteria API permits unsanitized literals when the **SELECT** or **GROUP_BY** parts of the SQL query contain a literal.

CHAPTER 8. FIXED ISSUES IN QUARKUS 1.7.6 GA

Quarkus 1.7.6 GA contains the following bug fixes:

8.1. SECURITY-RELATED BUG FIXES

- [CVE-2020-13956](#): A vulnerability was fixed in Quarkus 1.7.6 where the Apache **HttpClient** library versions earlier than 4.5.13 and 5.0.3 can misinterpret a malformed authority component in a request URIs that is passed to the library as a **java.net.URI** object and, as a result, pick the wrong target host for request execution.

8.2. CHANGES RELATED TO INTEGRATION WITH OTHER PRODUCTS AND TECHNOLOGIES

- [QUARKUS-278](#): Test and verify integration with Red Hat Single Sign-On version 7.4
- [QUARKUS-279](#): Introduce support for HTTP/2

8.3. DEVELOPER-EXPERIENCE-RELATED BUG FIXES

- [QUARKUS-449](#): Quarkus uses Protostream which uses shaded community versions of **javassist** and **protobuf-java**
- [QUARKUS-450](#): **JarResultBuildStep** - Duplicate entry META-INF/build.metadata entry warnings
- [QUARKUS-460](#): Quarkus gRPC fails with missing JSR305 on simple stub
- [QUARKUS-487](#): Maven compiler plugin 3.8.1 and Protostream - Marshaller JAVA files not compiled into class files
- [QUARKUS-525](#): Ensure that embedded fields with camel case names work in Spring Data JPA repositories
- [QUARKUS-526](#): Improved the parsing of the host and port in Eclipse Vert.x
- [QUARKUS-532](#): Fix **spring-data-jpa** field lookup with layered **@MappedSuperclasses**
- [QUARKUS-533](#): Add allow **java.lang** annotations for resources
- [QUARKUS-534](#): Fixed the default platform version range for the create command
- [QUARKUS-535](#): Fix Bootstrap dependencies
- [QUARKUS-537](#): Reduce error logging in Vert.x proxy address forwarding
- [QUARKUS-538](#): Remove the Bootstrap dependency from the BOM
- [QUARKUS-539](#): Add JSONP for cases where JSONB is not in scope
- [QUARKUS-540](#): Do not consider the hierarchy of **Object** when annotated with **@Valid**
- [QUARKUS-541](#): Ensure that gRPC work in native mode when used in an application with other Quarkus extensions

- [QUARKUS-544](#): Do not log warnings about duplicate **build.metadata** files
- [QUARKUS-545](#): Avoid copying sources directly into the source cache
- [QUARKUS-546](#): Improve Debugging experience by copying sources
- [QUARKUS-547](#): The **@Scope#scopeName** Spring annotation is now taken into account
- [QUARKUS-549](#): Fix the capability to connect to MySQL in XA mode
- [QUARKUS-551](#): Register String constructors for reflection
- [QUARKUS-557](#): Quarkus returns HTTP code **500 Internal Server Error** and a **NullPointerException** if it fails to parse an URI
- [QUARKUS-579](#): JWK does not support using chained certificates
- [QUARKUS-608](#): Remove spaces from **quarkus.container-image.group** when it is set match the username
- [QUARKUS-610](#): Improve logging and documentation about the **objcopy** dependency
- [QUARKUS-611](#): Fix the pause functionality in **SimpleScheduler**
- [QUARKUS-614](#): Do not log error message on bad request
- [QUARKUS-615](#): Create a new RESTEasy Context Map
- For Quarkus 1.7.6 GA an issue was fixed that caused the source cache directory structure to be incompatible with Mandrel. Sources were created in the **sources/src** subdirectory instead of the **sources/** subdirectory, so it was necessary to set the directory path to **sources:path/to/sources/src** when debugging a native executable using the GDB tool.

8.4. COMPONENT VERSION UPGRADES

- [QUARKUS-543](#): Upgrade Hibernate Validator version to 6.1.6.Final
- [QUARKUS-548](#): Upgrade Quarkus HTTP version to 3.0.15.Final
- [QUARKUS-606](#): Upgrade Eclipse Vert.x components to version 3.9.5
- [QUARKUS-625](#): Update Apache HTTP Client to version 4.5.13
- [QUARKUS-685](#): Update Infinispan version to 11.0.4 and Infinispan ProtoStream version to 4.3.4.Final

Revised on 2021-01-11 16:32:07 UTC