



Red Hat OpenShift Application Runtimes 1

RHOAR Thorntail Release Notes

For use with Red Hat OpenShift Application Runtimes Thorntail 2.4.0

Red Hat OpenShift Application Runtimes 1 RHOAR Thorntail Release Notes

For use with Red Hat OpenShift Application Runtimes Thorntail 2.4.0

Legal Notice

Copyright © 2019 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 Release Note contains important information related to Red Hat OpenShift Application Runtimes Thorntail 2.4.0

Table of Contents

PREFACE	3
CHAPTER 1. REQUIRED INFRASTRUCTURE COMPONENT VERSIONS	4
CHAPTER 2. SUPPORTED THORNTAIL RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS	5
CHAPTER 3. RELEASE COMPONENTS	6
3.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE	6
3.2. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE	6
3.3. ARTIFACTS REMOVED IN THIS RELEASE	6
3.4. ARTIFACTS DEPRECATED IN THIS RELEASE	6
CHAPTER 4. FEATURES	7
4.1. NEW AND CHANGED FEATURES	7
4.2. DEPRECATED FEATURES	7
4.3. TECHNOLOGY PREVIEW	8
CHAPTER 5. FIXED ISSUES	9
5.1. NOTABLE FIXED NON-SECURITY ISSUES	9
5.1.1. Introduce support for Maven 3.6	9
5.2. FIXED SECURITY ISSUES	9
CHAPTER 6. KNOWN ISSUES	10
6.1. MICROPROFILE FAULT TOLERANCE: CDI CONTEXTS AVAILABLE IN @TIMEOUT METHODS	10
6.2. MICROPROFILE METRICS: APPLICATION METRIC BEHAVIOR DOES NOT CONFORM TO METRICS SPECIFICATION	10
6.3. HARMLESS ERROR MESSAGE IN APPLICATION LOG: MISSING ORG.GLASSFISH:JAVAX.EL-API:3.0.1.B08-REDHAT-1	11
6.4. MICROPROFILE RESTCLIENT FAILS WHEN USING COMPLETIONSTAGE AND @PATHPARAM	12
CHAPTER 7. KNOWN ISSUES AFFECTING REQUIRED INFRASTRUCTURE COMPONENTS	13
CHAPTER 8. ADVISORIES RELATED TO THIS RELEASE	15

PREFACE

Date of release: 2019-04-17

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 THORNTAIL RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

The following resources define the supported configurations and integrations of Red Hat products with RHOAR Thorntail:

- For a list of technologies that are supported for integration with RHOAR Thorntail in production environments see the [Supported Thorntail 2.4.0 configurations and integrations](#) .
- For a list of RHOAR Thorntail runtime artifacts and their versions see the [Thorntail 2.4.0 component details page](#).

CHAPTER 3. RELEASE COMPONENTS

3.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE

The following supported artifacts have been introduced in this release.

- **io.thorntail:ee-security**
- **io.thorntail:jsonb**
- **io.thorntail:jaxrs-jsonb**

3.2. TECHNOLOGY PREVIEW ARTIFACTS INTRODUCED IN THIS RELEASE

No technology preview artifacts have been introduced in this release.

3.3. ARTIFACTS REMOVED IN THIS RELEASE

No artifacts have been removed in this release.

3.4. ARTIFACTS DEPRECATED IN THIS RELEASE

No artifacts have been declared deprecated in this release.

CHAPTER 4. FEATURES

4.1. NEW AND CHANGED FEATURES

Thorntail supports building applications with Java 11

Thorntail 2.4.0 introduces support for building and running application uberjars based on Java 11.



IMPORTANT

Note that Thorntail does not run on modulepath. Only classpath is supported. For additional information, see the *Java 11 support* section of the [Thorntail 2.4.0 community release notes](#).

Thorntail supports building applications with Java EE 8

Thorntail 2.4.0 introduces support for building and running application uberjars based on Java Enterprise Edition 8.

Thorntail implements MicroProfile 2.2

This release of RHOAR Thorntail implements the MicroProfile specification version 2.2.

Thorntail uses dependencies provided by EAP 7.2.0.GA

EAP dependencies used by RHOAR Thorntail have been updated and aligned with the 7.2.0.GA release of Red Hat JBoss Enterprise Application Platform.

Thorntail uses dependencies Red Hat SSO 7.3.0.GA

This RHOAR Thorntail release uses components provided by Red Hat Single Sign-On release version 7.3.0.GA.

4.2. DEPRECATED FEATURES



WARNING

The following deprecations might result in breaking changes to your existing applications using Thorntail. Ensure that you update your application configuration after upgrading from an earlier release of Thorntail.

@CreateSwarm annotation

The **@CreateSwarm** annotation is deprecated and should not be used anymore.

Configuration keys using the **swarm.*** prefix

Using the **swarm.*** prefix in the names of configuration keys is deprecated. It is still possible to use the **swarm.*** prefix, with the exception of:

- System properties passed to the Thorntail Maven plugin.
- Configuration properties injected in your application.

It is recommended to always use the **thorntail.*** prefix.

4.3. TECHNOLOGY PREVIEW

No technology preview features have been introduced in this release.

CHAPTER 5. FIXED ISSUES

This RHOAR Thorntail release incorporates all bugfixes from the upstream releases of Thorntail [2.2.1](#) and [2.3.0](#). Issues resolved in the community release are listed in the [Thorntail 2.4.0 Release Notes](#).

5.1. NOTABLE FIXED NON-SECURITY ISSUES

5.1.1. Introduce support for Maven 3.6

Description

An issue resulting in an **AbstractMethodError** when building Thorntail-based projects using Maven 3.6.0 has been resolved in this release. You can now build your Thorntail applications using Maven 3.6.0.

5.2. FIXED SECURITY ISSUES

- [CVE-2018-1067](#)
- [CVE-2018-1114](#)
- [CVE-2018-10862](#)
- [CVE-2018-10894](#)
- [CVE-2018-10912](#)
- [CVE-2018-11307](#)
- [CVE-2018-12022](#)
- [CVE-2018-12023](#)
- [CVE-2018-14718](#)
- [CVE-2018-14719](#)
- [CVE-2018-19360](#)
- [CVE-2018-19361](#)
- [CVE-2018-19362](#)
- [CVE-2018-1000180](#)

CHAPTER 6. KNOWN ISSUES

6.1. MICROPROFILE FAULT TOLERANCE: CDI CONTEXTS AVAILABLE IN @TIMEOUT METHODS

Description

If your application contains a **@Timeout** method that uses a contextual service, such as the **@RequestScoped MyService** shown in the example below, the contexts are not activated for that service.

```
@Inject
private MyService service;

@Timeout
public String doSomething() throws InterruptedException {
    return "Hello " + service.call();
}
```

The method is not **@Asynchronous** and should, therefore, be executed on the caller thread, which would make the Context and Dependency Injection (CDI) contexts available. However, the following debug message indicates that the contexts are not available:

```
2018-04-03 21:16:35,976 ERROR [io.undertow.request] (default task-1) UT005023: Exception
handling request to /: org.jboss.weld.context.ContextNotActiveException: WELD-001303: No active
contexts for scope type javax.enterprise.context.RequestScoped
```

Cause

This issue is caused by **@Timeout** methods always being invoked on a separate thread, even if they are not **@Asynchronous**.

Workaround

At the time of this release, there is no workaround available for this issue.

6.2. MICROPROFILE METRICS: APPLICATION METRIC BEHAVIOR DOES NOT CONFORM TO METRICS SPECIFICATION

Description

When you build and package your application and then run the resulting ***-thorntail.jar** uberjar, the application metric is not registered immediately upon deployment. The application metric is registered only after the monitored method is called.

For example, your Thorntail application contains a simple application metric, such as:

```
@ApplicationScoped
public class HelloService {
    @Counted(monotonic = true, name = "hello", absolute = true, displayName = "HELLO", description
= "Number of hello invocations")
    public String hello() {
```

```

    return "Hello from counted method";
  }
}

```



NOTE

You can test whether application metrics are registered by issuing an **OPTIONS** HTTP request to the **/metrics** REST endpoint. For example, **localhost:8080/metrics**, when running your application locally.

Cause

The implementation does not conform to the current version of the MicroProfile Metrics specification. The specification will likely change in the future to allow this behavior.

Workaround

At the time of this release, there is no workaround available for this issue.

6.3. HARMLESS ERROR MESSAGE IN APPLICATION LOG: MISSING ORG.GLASSFISH:JAVAX.EL-API:3.0.1.B08-REDHAT-1

Description

If your application, or any of its dependencies, depends on the Java Expression Language, it will display the following warning message during startup.

```

Failed downloading org/glassfish/javax.el-api/3.0.1.b08-redhat-1/javax.el-api-3.0.1.b08-redhat-1.pom
from https://repository.jboss.org/nexus/content/groups/public/. Reason:
org.eclipse.aether.transfer.ArtifactNotFoundException: Could not find artifact org.glassfish:javax.el-
api:pom:3.0.1.b08-redhat-1 in jboss-public-repository-group
(https://repository.jboss.org/nexus/content/groups/public/)
Failed downloading org/glassfish/javax.el-api/3.0.1.b08-redhat-1/javax.el-api-3.0.1.b08-redhat-1.pom
from http://repo.gradle.org/gradle/libs-releases-local/. Reason:
org.eclipse.aether.transfer.ArtifactNotFoundException: Could not find artifact org.glassfish:javax.el-
api:pom:3.0.1.b08-redhat-1 in gradle (http://repo.gradle.org/gradle/libs-releases-local)
Failed downloading org/glassfish/javax.el-api/3.0.1.b08-redhat-1/javax.el-api-3.0.1.b08-redhat-1.pom
from https://repo.maven.apache.org/maven2/. Reason:
org.eclipse.aether.transfer.ArtifactNotFoundException: Could not find artifact org.glassfish:javax.el-
api:pom:3.0.1.b08-redhat-1 in central (https://repo.maven.apache.org/maven2)
Failed downloading org/glassfish/javax.el-api/3.0.1.b08-redhat-1/javax.el-api-3.0.1.b08-redhat-1.pom
from http://repo1.maven.org/maven2/. Reason:
org.eclipse.aether.transfer.ArtifactNotFoundException: Could not find artifact org.glassfish:javax.el-
api:pom:3.0.1.b08-redhat-1 in central (http://repo1.maven.org/maven2)

```

The message is harmless and does not impact the functionality of the application.

Cause

The likely cause of this issue is related to the way dependency resolution works in Thorntail. During the dependency resolution phase, Thorntail ignores dependency exclusions, and thus pulls in **javax.el-api**, despite **javax.el-api** being excluded in the EAP BOM. Because it is interpreted as a valid dependency, it is indicated as missing due to being absent from the repository, which causes the error messages displayed in the build log.

Workaround

At the time of this release, there is no workaround available for this issue.

6.4. MICROPROFILE RESTCLIENT FAILS WHEN USING COMPLETIONSTAGE AND @PATHPARAM

Description

Using the asynchronous invocation feature of MicroProfile RestClient can result in an error.

An example request containing parameters

```
@Path("/api/hello")
@RequestScoped
@RegisterRestClient
public interface HelloService {

    @GET
    @Path("/{name}")
    @Produces(TEXT_PLAIN)
    @Consumes(TEXT_PLAIN)
    CompletionStage<String> helloAsync(@PathParam("name") String name);

}
```

Invoking a MicroProfile RestClient interface method that returns **CompletionStage** and has a **@PathParam** parameter results in the following error:

```
RESTEASY003720: path param name has not been provided by the parameter map
```

No workaround is available for this issue at the time of this release.

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 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 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.

- [RHSA-2019:0877](#)