Red Hat build of Thorntail 2.5

Release Notes for Thorntail 2.5

For use with Thorntail 2.5.1

Last Updated: 2020-07-01

Red Hat build of Thorntail 2.5 Release Notes for Thorntail 2.5

For use with Thorntail 2.5.1

Legal Notice

Abstract

This Release Note contains important information related to Thorntail 2.5.1

Table of Contents

PREFACE

Date of release: 2020-05-12

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.

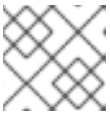
Prerequisites

- 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.

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
Fabric8 Maven Plugin	4.3.1
JDK[a][b]	OpenJDK 8, OpenJDK 11[c]
Red Hat Enterprise Linux 7[d]	7.7
Red Hat Enterprise Linux 8[e]	8.1
OpenShift Container Platform (OCP)[f]	3.11, 4.3
Minishift	1.34.2 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] 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 THORNTAIL RUNTIME COMPONENT CONFIGURATIONS AND INTEGRATIONS

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

- For a list of technologies that are supported for integration with Thorntail in production environments see the Supported Thorntail 2.5.1 configurations and integrations .
- For a list of Thorntail runtime artifacts and their versions see the Thorntail 2.5.1 component details page.

CHAPTER 3. RELEASE COMPONENTS

3.1. SUPPORTED ARTIFACTS INTRODUCED IN THIS RELEASE

The following supported artifacts are introduced in this release:

- **io.thorntail:ejb-mdb**
- **org.jboss.resteasy:resteasy-client-microprofile**

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

The following artifact is removed in this release.

- **io.smallrye:smallrye-rest-client**

3.4. ARTIFACTS DEPRECATED IN THIS RELEASE

No artifacts have been declared deprecated in this release.

CHAPTER 4. FEATURES

4.1. NEW FEATURES AND FEATURE UPGRADES

4.1.1. Support for Thorntail Runtime on IBM Z

The Red Hat build of Thorntail for s390x platform is supported only in OpenShift environments provisioned on IBM Z infrastructure. Running an Thorntail application on a stand-alone installation of RHEL on IBM Z is not supported.

Eclipse OpenJ9 Java images for IBM Z and new images for products supported on IBM Z are available in the Red Hat Ecosystem Catalog .

4.1.2. Deploying example applications on OpenShift provisioned on IBM Z infrastructure

To deploy the example applications on OpenShift environments provisioned on IBM Z infrastructure, specify the relevant IBM Z image name in the **pom.xml** file and commands.

Some of the example applications also require other products, such as Red Hat Data Grid to demonstrate the workflows. In this case, you must also change the image names of these products to their relevant IBM Z image names in the YAML file of the example applications.

The Secured example application in Thorntail requires Red Hat SSO 7.3. Since Red Hat SSO 7.3 is not supported on IBM Z, the Secured example is not available for IBM Z.

4.1.3. Feature upgrades

This release of Thorntail introduces the following feature upgrades:

Eclipse MicroProfile 3.0 support

This release implements Eclipse MicroProfile 3.0 by including the latest SmallRye artifacts. The following specifications have been upgraded to the versions listed below:

- Health Check 2.0
  - This release introduces the **@Liveness** and **@Readiness** annotations, replacing the **@Health** annotation that is now deprecated (see also the *Changed and deprecated features* section below).
  - The **/health/live** and **/health/ready** endpoints are introduced in this release. The **/health** endpoint now serves as the unified endpoint for the **@Health**, **@Liveness** and **@Readiness** checks.
- Metrics 2.0
  - The **@ConcurrentGauge** annotation is introduced in this release, as a replacement of the **@Counter(monotonic = false)** annotation (see also the *Changed and deprecated features* section below).
- Rest Client 1.3
  - The Rest Client interfaces can now extend **Closeable** and **AutoCloseable** interfaces.
  - The **RestClientBuilder** now includes methods for SSL configuration.

Eclipse MicroProfile JWT improvements

The **microprofile-jwt** fraction has received a number of improvements in this release. The changes are backwards compatible and not expected to break your application code.

- The **@LoginConfig** annotation is no longer required, if you set the **thorntail.microprofile.jwt.token.realm** configuration property in your application. If you use the **@LoginConfig** annotation with the **thorntail.microprofile.jwt.token.realm** property set, ensure that the value of the **realmName** property of the **@LoginConfig** annotation matches the of the **thorntail.microprofile.jwt.token.realm** configuration property.

- You can set **thorntail.microprofile.jwt.enable** configuration property value to **false** to disable the Eclipse MicroProfile JWT authentication mechanism. The default value of the **thorntail.microprofile.jwt.enable** configuration property is **true**.

- You must use the **thorntail.microprofile.jwt.token.signer-pub-key-location** to specify external key assets. Previously available alternate methods are deprecated. (For details, see *Changed and deprecated features* below.)

- The **thorntail.microprofile.jwt.path.groups** configuration property can be used to select an arbitrary claim inside a JWT token to function as the **groups** claim expected by Eclipse MicroProfile JWT. This is useful if no **groups** claim is present in the JWT token, but some other claim exists that contains the groups information. Unless the custom claim is a standard **scope** claim, its value must be formatted as an array of strings. Use the following syntax to reference a claim: **path/to/claim**.

- You can use the **thorntail.microprofile.jwt.claims.groups** configuration property to provide a default value for the **groups** claim if the JWT token contains no groups information at all.

- You can use the **thorntail.microprofile.jwt.token.roles.map** configuration property to inline the roles configuration in the Thorntail YAML configuration.

ejb-mdb fraction introduced

This release introduces the **ejb-mdb** fraction. This fraction enables you to deploy Message-Driven Beans, but it does not provide all the components required to set up a messaging infrastructure for your application. This means that you can not use this fraction to create a messaging server, but you can use it connect to an existing external messaging server, if you also provide the correct resource adapter.

4.1.4. Updated components

The components updated in this Thorntail release are:

Red Hat JBoss EAP 7.2.7.GA

The EAP dependencies in Thorntail have been updated to Red Hat JBoss Enterprise Application Platform 7.2.7.GA release.

Red Hat SSO 7.3.7.GA

The single sign-on components in Thorntail have been updated to Red Hat Single Sign-On 7.3.7.GA release.

Jaeger 0.34.1

This release of Thorntail contains Jaeger 0.34.1 client components.

Apache Thrift 0.13.0

This release of Thorntail contains Apache Thrift 0.13.0 components.

RESTEasy 3.9.3.SP1

This release of Thorntail contains RESTEasy 3.9.3.SP1 framework.

Jackson 2.10.2

This release of Thorntail contains Jackson 2.10.2 libraries.

SmallRye Config 1.3.6.SP01

This release of Thorntail contains SmallRye Config 1.3.6.SP01.

4.2. CHANGED AND DEPRECATED FEATURES

Microprofile 3.0

- Health Check 2.0
  - The **@Health** annotation is now deprecated. Use the **@Liveness** and **@Readiness** annotations instead.

    CAUTION

    This is a breaking change. Ensure that you update your application code to avoid encountering issues with your applications after upgrading to the latest release of Thorntail.

  - The **outcome** and **state** fields in the JSON response format are replaced by the **status** field.

    CAUTION

    This is a breaking change. Ensure that you update your application code and external monitoring to avoid encountering issues with your applications after upgrading to the latest release of Thorntail.

- Metrics 2.0
  - Some of the base metrics (defined by the Eclipse MicroProfile Metrics 2.0 specification) changed their type from **Counter** to **Gauge**.
  - Some of the base metrics as well as some of the vendor metrics (provided by Thorntail) were renamed. Notably, the names of all accumulating counters now contain the suffix **total**.
  - In the OpenMetrics output format (formerly known as the Prometheus format), the metric scope and name are now separated by an underscore (_) symbol. This replaces the colon (**:**) symbol used as the separator in previous releases.
  - Metrics are no longer identified by their **String** name, but by a **MetricID** class. The **MetricID** consists of a **String** name and a set of key-value pairs called tags. In application code, this set of tags is represented as **Map<String, String>**.
  - The **@Counted** annotation no longer includes the **monotonic** parameter. Previously, the default value of the **monotonic** parameter was **false**. Now, the **@Counter** annotation always behaves as if the **monotonic** parameter was set to **true**. When you want your annotation to behave as when **monotonic** is set to **false**, use the **@ConcurrentGauge** annotation instead.

CAUTION

These are breaking changes. Ensure that you update your application code and external monitoring configuration to avoid encountering issues with your applications after upgrading to the latest release of Thorntail.

- Rest Client 1.3
  - In this release, Thorntail replaces the Eclipse MicroProfile Rest Client implementation from the SmallRye project, with the Eclipse MicroProfile Rest Client implementation from RESTEasy. As a result of this change, all public APIs and SPIs that were previously present in the **io.smallrye.restclient** package, such as **RestClientProxy**, are now part of the **org.jboss.resteasy.microprofile.client** package. Ensure that you use the new package name in your code and logging configuration.
  - The specification now mandates that **application/json** is used as a default media type, if the Rest Client interface does not include the **@Produces**/**@Consumes** annotations. If you do not use the **@Produces**/**@Consumes** annotations in your client interfaces, this change might break your application.

CAUTION

These are breaking changes. Ensure that you update your application code and logging configuration to avoid encountering issues with your applications after upgrading to the latest release of Thorntail.

Eclipse MicroProfile JWT

When specifying external key assets, the following referencing methods are now deprecated:

- Using the **file:** and **classpath:** prefixes in the value of the **thorntail.microprofile.jwt.token.signer-pub-key** configuration property to point to the location of an external key asset.
- Using the **thorntail.microprofile.jwt.token.jwks-uri** configuration property to refer to an external JWK Set.

Instead, use **thorntail.microprofile.jwt.token.signer-pub-key-location** configuration property.

Using configuration properties in the **thorntail.microprofile.jwtauth** namespace is now deprecated. While this is not a breaking change, you are encouraged to update your configuration to use properties from the **thorntail.microprofile.jwt** namespace.

The wildfly-swarm.useUberJar system property

The **wildfly-swarm.useUberJar** system property previously used by the Thorntail Maven plugin is no longer recognized. To ensure that the Thorntail Maven plugin runs in uberjar mode (as opposed to classpath mode), use the **thorntail.useUberJar** system property.

CAUTION

This is a breaking change. Ensure that you use the new name when running Maven to avoid encountering issues with your applications after upgrading to the latest release of Thorntail.

4.3. TECHNOLOGY PREVIEW

No technology preview features have been introduced in this release.

CHAPTER 5. FIXED ISSUES

This Thorntail release contains the following bug fixes.

5.1. HTTP/2 PROTOCOL IS AUTOMATICALLY ENABLED ON IBM JRE

Description

Prior to this release, HTTP/2 protocol was not automatically enabled on IBM JRE. This issue is fixed and HTTP/2 is automatically enabled on IBM JRE when the required cipher suite is available.

5.2. ALL CONFIGURATIONS WORK AS EXPECTED ON IBM JRE

Prior to this release, configurations of some elements, such as security realms, were not applied on IBM JRE. For example, the following YAML file did not load correctly.

```
thorntail:
  management:
    http-interface-management-interface:
      security-realm: ManagementRealm
    security-realms:
      ManagementRealm:
        in-memory-authentication:
          users:
            albert:
              password: wow_E=m*c^2
        in-memory-authorization:
          users:
            albert:
              roles:
              - admin
```

This issue occurred because the Thorntail configuration system expected Class.getMethods() to return methods in a certain order. However, by definition, the method Class.getMethods() does not return methods in any particular order.

The issue is fixed. Thorntail explicitly sorts the methods that are returned and the configurations work as expected on IBM JRE.

5.3. NOTABLE FIXED NON-SECURITY ISSUES

5.3.1. Eclipse MicroProfile Metrics: Application metric behavior does not conform to metrics specification

Description

Prior to this release, metrics would not be registered upon deployment, but would instead be registered when the metric method was first called. This issue was fixed and metrics are now registered correctly.

5.3.2. Eclipse MicroProfile RestClient fails when using `CompletionStage` and `@PathParam`

Description

Prior to this release, invoking a Eclipse MicroProfile RestClient interface method that returns **CompletionStage** and has a **@PathParam** parameters would result in an error. This issue was fixed and asynchronous Eclipse MicroProfile RestClient invocations now work correctly.

5.3.3. SmallRye Config: Incorrect interpretation of escaped backslash ('\\') character sequence

Description

Due to a bug in the SmallRye Config component in the previous release, it was not possible to use a backslash character to escape another backslash character. This could even result in wrong array conversion, if there was a comma character right after the double backslash sequence. See the example below for details. This issue has been resolved in this release and the behavior no longer occurs.

Example

```
thorntail:
  microprofile:
    config:
      config-sources:
        propertiesSource:
          properties:
            arrayProperty: element1,element2\\,element3,element41\,element42,ele\\ment5
```

The expected interpretation of the value entered in the **arrayProperty** in the example above is: **element1**, **element2**, **element3**, **element41,element42**, **ele\ment5**

Instead, the **arrayProperty** value entered as shown in the example above was interpreted as: **element1**, **element2\,element3**, **element41\,element42**, **ele\\ment5**

5.3.4. Thorntail Topology OpenShift fraction does not work on OpenShift 4.2

Description

Prior to this release, using the Thorntail Topology fraction on an OpenShift 4.2 cluster resulted in the following exception at runtime:

```
org.jboss.msc.service.StartException in service "swarm.topology.openshift".service-watcher: Failed to start service
...
Caused by: com.openshift.restclient.authorization.ResourceForbiddenException: forbidden: User "system:anonymous" cannot get path "/apis" forbidden: User "system:anonymous" cannot get path "/apis"
```

The behavior was caused by an issue with the OpenShift Java REST Client. The issue has been resolved in the 9.0 release of the OpenShift Java REST Client, and Thorntail now includes the new version.

5.4. FIXED SECURITY ISSUES

For a list of resolved security issues, see *Advisories related to this release*.

CHAPTER 6. KNOWN ISSUES

6.1. THORNTAIL APPLICATIONS FAIL TO BOOT DUE TO LOGGING ISSUES

Description

Thorntail is based on WildFly. In Wildfly, the value of the **java.util.logging.manager** property should always be set to **org.jboss.logmanager.LogManager** (JBoss LogManager). The JBoss Modules initialize the log manager while booting. However, in some cases, **java.util.logging** is called before the log manager is set. In such cases, the Thorntail application fails to boot and returns the following error:

```
ERROR: WFLYCTL0013: Operation ("parallel-extension-add") failed - address: ([])
java.lang.RuntimeException: WFLYCTL0079: Failed initializing module org.jboss.as.logging
        ...
Caused by: java.util.concurrent.ExecutionException: java.lang.IllegalStateException:
WFLYLOG0078: The logging subsystem requires the log manager to be
org.jboss.logmanager.LogManager.
The subsystem has not be initialized and cannot be used.
To use JBoss Log Manager you must add the system property "java.util.logging.manager" and set it
to "org.jboss.logmanager.LogManager"
```

Cause

The Thorntail application fails to boot if **java.util.logging** is initialized too early. Some of the reasons for **java.util.logging** being initialized too early are:

- The application uses Java agents, such as the Jolokia agent
- The JDK itself uses logging

Workaround

To use Thorntail on OpenShift, it is recommended to switch off Java agents that are available in the Red Hat Java S2I images.

You can switch off the agents using one of the following ways:

- To switch off Jolokia agent, you should set the following environment variables:
  - **AB_OFF** to **true**
  - **AB_JOLOKIA_OFF** to **true**

The following example shows an OpenShift deployment configuration where the environment variables are set.

```
apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  ...
spec:
  ...
  template:
    metadata:
```

```
    ...
  spec:
    containers:
    - image: ...
      env:
      - name: AB_JOLOKIA_OFF
        value: "true"
      - name: AB_OFF
        value: "true"
      ...
```

- You can also use Fabric8 Maven plugin to generate the following YAML files. These files set the environment variables as shown in the following code.

```
<plugin>
  <groupId>io.fabric8</groupId>
  <artifactId>fabric8-maven-plugin</artifactId>
  <version>${version.io.fabric8.fabric8-maven-plugin}</version>
  <executions>
    <execution>
      <goals>
        <goal>resource</goal>
        <goal>build</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <resources>
      <env>
        <AB_OFF>true</AB_OFF>
        <AB_JOLOKIA_OFF>true</AB_JOLOKIA_OFF>
      </env>
    </resources>
  </configuration>
</plugin>
```

- You can explicitly configure JBoss LogManager as the **java.util.logging.manager**. The following example shows you the configuration:

```
java -Xbootclasspath/p:/home/test/.m2/repository/org/jboss/logmanager/jboss-
logmanager/2.1.14.Final-redhat-00001/jboss-logmanager-2.1.14.Final-redhat-
00001.jar:/home/test/.m2/repository/org/wildfly/common/wildfly-common/1.5.1.Final-redhat-
00001/wildfly-common-1.5.1.Final-redhat-00001.jar -
Djboss.modules.system.pkgs=org.jboss.logmanager,org.wildfly.common -
Djava.util.logging.manager=org.jboss.logmanager.LogManager -jar myapp-thorntail.jar
```

6.2. ECLIPSE 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.

6.3. 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 form 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. 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
```

6.5. THORNTAIL ARQUILLIAN ADAPTER IGNORES MVN -S SETTINGS.XML

Description

When attempting to pass additional repositories referenced in the **setting.xml** file to the Thorntail Arquillian adapter when executing integration tests and unit tests, the **settings.xml** file is not recognized and the additional repositories are not configured. This issue results in build failure due to missing artifacts, which, in turn, causes the tests to fail.

Workaround

To avoid this issue, manually edit the **<configuration>** sections under the Maven Surefire and/or Maven

Failsafe plugin entries in the **pom.xml** file of your Maven project, manually specifying the **<org.apache.maven.user-settings>${session.request.userSettingsFile.path}</org.apache.maven.user-settings>** property to be exported when testing your application. See example for details:

pom.xml

```
<project>
  ...
  <build>
    <plugins>
      ...
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-failsafe-plugin</artifactId>
        <configuration>
          <org.apache.maven.user-
settings>${session.request.userSettingsFile.path}</org.apache.maven.user-settings>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <configuration>
          <systemPropertyVariables>
            <org.apache.maven.user-
settings>${session.request.userSettingsFile.path}</org.apache.maven.user-settings>
          </systemPropertyVariables>
        </configuration>
      </plugin>
      ...
    </plugins>
  </build>
  ...
</project>
```

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.

- RHSA-2020:2067