Red Hat JBoss Enterprise Application Platform 8.0

Release notes for Red Hat JBoss Enterprise Application Platform 8.0

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform release 8.0

Last Updated: 2024-02-27

Red Hat JBoss Enterprise Application Platform 8.0 Release notes for Red Hat JBoss Enterprise Application Platform 8.0

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform release 8.0

Legal Notice

Abstract

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform release 8.0.

Table of Contents

PREFACE

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform 8.0.

PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

Procedure

1. Click the following link to **create a ticket**.
2. Enter a brief description of the issue in the **Summary**.
3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.
4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

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. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our CTO Chris Wright's message.

CHAPTER 1. HOW TO READ THE RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 8.0 DOCUMENTATION

We are in the process of modernizing the Red Hat JBoss Enterprise Application Platform 8.0 documentation. We are working to create more solution-centric documentation.

The JBoss EAP 8.0 documentation contains content specific to the JBoss EAP 8.0 release including new and enhanced features found in JBoss EAP 8.0. Functionality from previous releases that are still supported in JBoss EAP 8.0 can be accessed in the JBoss EAP 7.4 documentation set. You can access the documentation set at Product Documentation for Red Hat JBoss Enterprise Application Platform 7.4.

The following is a suggested approach for using the JBoss EAP 8.0 documentation:

1. Read the JBoss EAP 8.0 Release notes to learn about new, enhanced, unsupported, and removed features.

2. Read the other JBoss EAP 8.0 documentation set for detailed information about new and enhanced features.

3. Read the JBoss EAP 8.0 Migration Guide for details on how to migrate applications to JBoss EAP 8.0.

4. If you need information on features supported from previous releases that have not been enhanced in JBoss EAP 8.0, see the JBoss EAP 7.4 documentation set at Product Documentation for Red Hat JBoss Enterprise Application Platform 7.4. For example, development and configuration guides are available in the JBoss EAP 7.4 documentation set.

CHAPTER 2. SUPPORTED CONFIGURATIONS

For more information about supported and tested configurations for Java Virtual Machines (JVMs) and JBoss EAP 8.0, including commonly used operating systems, databases, and JMS brokers, see the Red Hat JBoss Enterprise Application Platform 8.0 supported configurations Knowledgebase article on the Red Hat Customer Portal.

For more information about included modules, supported standards, and Red Hat JBoss Enterprise Application Platform component details, see the following support documents:

- Red Hat JBoss Enterprise Application Platform (EAP) 8 Included Modules
- Red Hat JBoss Enterprise Application Platform Supported Standards
- Red Hat JBoss Enterprise Application Platform Component Details

OpenShift images

Builder and Runtime images are supported for OpenJDK 17 / RHEL 8 on Intel, IBM systems Z & P, and ARM architectures.

Red Hat build of Keycloak SAML adapters support

You can now install Red Hat build of Keycloak SAML adapters using the Keycloak SAML adapter Galleon feature pack. For information about Red Hat build of Keycloak, see the Red Hat build of Keycloak product page.

MariaDB Driver logging dependency

In JBoss EAP 8.0, the MariaDB Driver includes a dependency on the **org.slf4j** module to improve logging functionality. This prevents possible errors when **slf4j** classes are unavailable, and ensures proper handling of log messages. When deploying the MariaDB Driver module, you must ensure the **slf4j** dependencies are available to leverage this enhancement. For more information, see How to configure MariaDB driver 3.0+ as a JBoss Module in EAP 7 / 8.

JBoss EAP 8.0 Operator

JBoss EAP 8.0 now supports the EAP operator. You can now deploy your JBoss EAP 8.0 applications using the JBoss EAP operator. For more information, see JBoss EAP Operator Support Policy.

CHAPTER 3. NEW FEATURES AND ENHANCEMENTS

JBoss EAP 8.0 introduces the following new features and enhancements.

3.1. JAKARTA EE 10 SUPPORT

JBoss EAP 8 provides support for Jakarta EE 10 and implements the Jakarta EE 10 Core Profile, Web Profile and Full Platform standards, including:

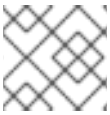
- Jakarta Activation 2.1
- Jakarta Annotations 2.1
- Jakarta Authentication 3.0
- Jakarta Authorization 2.1
- Jakarta Batch 2.1
- Jakarta Bean Validation 3.0
- Jakarta Concurrency 3.0
- Jakarta Connectors 2.1
- Jakarta Contexts and Dependency Injection 4.0
- Jakarta Debugging Support for Other Languages 2.0
- Jakarta Dependency Injection 2.0
- Jakarta Enterprise Beans 4.0
- Jakarta Enterprise Web Services 2.0
- Jakarta Expression Language 5.0
- Jakarta Interceptors 2.1
- Jakarta JSON Binding 3.0
- Jakarta JSON Processing 2.1
- Jakarta Mail 2.1
- Jakarta Messaging 3.1
- Jakarta Persistence 3.1
- Jakarta RESTful Web Services 3.1
- Jakarta Security 3.0
- Jakarta Server Faces 4.0
- Jakarta Server Pages 3.1

- Jakarta Servlet 6.0
- Jakarta SOAP with Attachments 1.3
- Jakarta Standard Tag Library 3.0
- Jakarta Transactions 2.0
- Jakarta WebSocket 2.1
- Jakarta XML Binding 4.0
- Jakarta XML Web Services 4.0

Jakarta EE 10 has many changes when compared to Jakarta EE 8. For more information, see How to migrate your JBoss EAP applications from Jakarta EE 8 to Jakarta EE 10.

Package Namespace Change

The packages used for all EE APIs have changed from **javax** to **jakarta**. This follows the move of Java EE to the Eclipse Foundation and the establishment of Jakarta EE.



NOTE

This change does not affect **javax** packages that are part of Java SE.

Additional resources

- For more information, see The javax to jakarta Package Namespace Change .

3.2. RED HAT INSIGHTS JAVA CLIENT

JBoss EAP 8.0 version onward contains the Red Hat Insights Java client. The Red Hat Insights Java client is enabled for JBoss EAP only if JBoss EAP is installed on Red Hat Enterprise Linux (RHEL), and the RHEL system has Red Hat Insights client installed, configured, and registered. For more information, see the Client Configuration Guide for Red Hat Insights .

The Red Hat Insights dashboard for Runtimes will be available in a future release on Red Hat Hybrid Cloud Console. Similar to the RHEL dashboard which is available on the Red Hat Hybrid Cloud Console, the Runtimes dashboard will show the inventory of the Runtimes installations, CVE details, and help you select the JVM options.

You can opt-out of the Red Hat Insights client by setting the environment variable **RHT_INSIGHTS_JAVA_OPT_OUT** to **true**. For more information, see the knowledge base article Red Hat Insights for Runtimes.

3.3. MANAGEMENT CONSOLE

Inclusive language, label changes

Toward Red Hat's commitment to replacing problematic language in our code, documentation, and web properties, beginning with 8.0, the JBoss EAP management console will display more inclusive wording and labels. Specifically, you will notice the following changes to the management console resource addresses and user interface elements:

New term	Previous term
primary	master
secondary	slave
blocklist	blacklist
allowlist	whitelist

Adding, editing, and removing constant HTTP headers to response messages

In the JBoss EAP 8.0 management console, you can now add, edit, or remove constant HTTP response headers. To add a new path and header, from the Server page, select **Constant Headers**, then click **Add**. To edit or remove an existing path header, select the path whose header you want to modify, then click either **Edit** or **Remove**.

Displaying Java Message Service bridge statistics for processed messages

A message bridge consumes messages from a source queue or topic, then sends them on to a target queue or topic, usually on a different server. A bridge can also send messages from one cluster to another. The Java Message Service (JMS) bridge provides statistics about messages that the bridge processed. Specifically, it collects the following data:

- number of messages successfully committed (message count)
- number of messages aborted (messages aborted)

With this update, the JBoss EAP 8.0 management console includes a new **JMS Bridge** column to display these statistics in the Runtime section. Note that this new feature affects the **/subsystem=messaging-activemq/jms-bridge=*** resource.

Configuring enhanced audit logging

In the JBoss EAP 8.0 management console, you can configure the following two additional audit logging attributes in your **/subsystem=elytron/syslog-audit-log=*** resource:

- **syslog-format**
  Define the format for your audit log messages. Supported values are **RFC3164** and **RFC5424**. ("RFC" stands for "request for comments.")
- **reconnect-attempts**
  Define the maximum number of failed attempts JBoss EAP should make to connect to the syslog server before closing the endpoint.

/deployment subresources require include-runtime=true

With Red Hat JBoss Enterprise Application Platform 8.0, the submodel of **/deployment** has changed to runtime. For management operations that use **/deployment** subresources you must add **include-runtime=true**.

Starting servers in suspended mode

You can now use the JBoss EAP 8.0 management console to start servers in suspended mode. Select the new **Start in suspended mode**option, available in the following drop-down menus:

- **Runtime > Topology**

- **Runtime > Server Groups**
- **Runtime > Server Groups > Server**
- **Runtime > Host > Server**

Configuring the certificate-authority attribute for the certificate-authority-account resource

With JBoss EAP 8.0, you can use any certificate authority for your **certificate-authority-account** Elytron resource. Previously, JBoss EAP supported only the Let's Encrypt certificate authority, and the **certificate-authority** attribute was not configurable.

With this update, you can add, configure, or remove any certificate authority by opening the JBoss EAP management console and clicking **Configuration > Subsystems > Security > Other Settings > Other Settings > Certificate Authority**. From there, click **Add** to add a new certificate authority. To modify one you already have, select it, then click **Edit**. To remove a certificate authority, select it, then click **Remove**.

Configuring the OCSP as an Elytron trust manager

With JBoss EAP 8.0, you can configure the Online Certificate Status Protocol (OCSP) as the trust manager for the Elytron **undertow** subsystem. Previously, JBoss EAP supported only a certificate revocation list (CRL) as trust manager.

With this update, you can configure the OCSP as your trust manager by opening the JBoss EAP management console and clicking **Configuration > Subsystems > Elytron > Other Settings > SSL > Trust Manager**. Next, either select or create a trust manager and then, from the Trust Manager window, select the **OCSP** tab and click **Add**.

Pausing Java Message Service topics

From the JBoss EAP 8.0 management console, you can now navigate to **Runtime > Messaging > Server > Server Name > Destination** to select and then pause a Java Message Service (JMS) topic. After you address the related messaging issue, you can also resume the paused topic. JMS previously sent messages to all active subscribers without any way to interrupt them.

Non-heap memory usage added to server status preview

With JBoss EAP 8.0, you can see more information in the server status preview about the memory consumption of your server. Previously, the preview displayed only heap memory usage: **Used** and **Committed**. With this update, it also displays the same information for non-heap memory usage.

Automatically add or update credential store passwords when you add or update a datasource

Beginning with JBoss EAP 8.0, when you create a datasource from the management console, you can automatically add a password for that datasource to your credential store. From the management console, select **Configuration > Subsystems > Datasources**, then click **Add** to add a new datasource. Next, enter the credential store name where you want to save the password for the new datasource, an alias for the credential, and the plain text password you want to use. To modify an existing datasource, select it, then click **Edit**.

Create, read, update, and delete Elytron resources

From the JBoss EAP 8.0 management console, you can now create, read, update, or delete any of the following four evidence decoders:

- Aggregate Evidence Decoders
- Custom Evidence Decoders

- X500 Subject Evidence Decoders
- X509 Subject Alt Name Evidence Decoder

To take one of these actions, navigate to **Configuration > Subsystems > Security > Mappers & Decoders > Evidence Decoder**.

Viewing the deployment hash value

The JBoss EAP 8.0 management console can now display your deployment hash value in the deployment preview. This means that you can determine at a glance whether your deployment was valid and successful.

Adding and configuring interceptors in the EJB 3 subsystem

From the JBoss EAP 8.0 management console, you can now add and configure system-wide, server-side interceptors in the **ejb3** subsystem. From the console, select **Configuration > EJB > Container** to make your additions or changes.

Configuring Infinispan distributed web session affinity

With JBoss EAP 8.0, in the **distributable-web** subsystem, you now have more control over the affinity, or load balancer "stickiness", of a distributed web session. To change your session affinity to something other than the **Primary-owner** default, in the management console, click **Configuration > Distributable Web > View > Infinispan Session**. Next, choose a session and select **Affinity** to make your changes. Affinity options now include the following:

- **Local**
- **None**
- **Primary-owner**
- **Ranked**

Previously, the only available affinity was **Primary-owner**.

Configuring global directories in EE subsystem

With the JBoss EAP 8.0 management console, you can now configure a new **ee** subsystem resource, **/subsystem=ee/global-directory=***. You can use a global directory to add content to a deployment class path without listing the contents of the directory. To configure a global directory resource, navigate to **Configuration > Subsystems > EE > Globals**

Configuring cipher suites in Elytron

With the JBoss EAP 8.0 management console, you can now enable TLS 1.3 cipher suites using the **cipher-suite-names** attribute to secure your network connection. Specifically, you can now configure the following **elytron** subsystem resources:

- **/subsystem=elytron/client-ssl-context=***
- **/subsystem=elytron/server-ssl-context=***

To configure the **cipher-suite-names** attribute for the **/subsystem=elytron/client-ssl-context=*** resource from the management console, navigate to **Configuration > Subsystems > Security > Other Settings > SSL > Client SSL Context**.

To configure the **cipher-suite-names** attribute for the **/subsystem=elytron/server-ssl-context=*** resource from the management console, navigate to **Configuration > Subsystems > Security > Other Settings > SSL > Server SSL Context**.

Securing applications and management console with OIDC

With the JBoss EAP 8.0, you can secure applications deployed to JBoss EAP, and the JBoss EAP management console with OpenID Connect (OIDC) from the management console. JBoss EAP 8.0 provides native support for OpenID Connect (OIDC) with the **elytron-oidc-client** subsystem.

To configure the **elytron-oidc-client** subsystem from the management console, navigate to **Configuration > Subsystems > Elytron OIDC Client**

To secure applications deployed to JBoss EAP, configure the following resources:

- **provider**
- **secure-deployment**

For more information, see Securing applications with OIDC in the *Using single sign-on with JBoss EAP* guide.

To secure the JBoss EAP management interfaces, configure the following resources:

- **provider**
- **secure-deployment**
- **secure-server**

Additionally, you can configure role-based access control (RBAC) for management console when securing it with OIDC by navigating to **Access Control** and clicking **Enable RBAC**.

For more information, see Securing the JBoss EAP management console with an OpenID provider in the *Using single sign-on with JBoss EAP* guide.

NOTE

You can use the **realm** resource to configure a Red Hat build of Keycloak realm. This is provided for convenience. You can copy the configuration in the keycloak client adapter and use it in the **realm** resource configuration. However, using the **provider** resource is recommended instead.

3.4. MANAGEMENT CLI

Registering web context when deploying an application

You can use the **deployment deploy-file** command from the management command-line interface (CLI) to deploy applications to a standalone server or in a managed domain.

Deploy an application to a standalone server

```
deployment deploy-file /path/to/test-application.war
```

Deploy an application to all server groups in a managed domain

```
deployment deploy-file /path/to/test-application.war --all-server-groups
```

Deploy an application to specific server groups in a managed domain

```
deployment deploy-file /path/to/test-application.war --server-groups=main-server-group,other-server-group
```

In the preceding examples, the default value for the **runtime-name** attribute is **test-application.war**.

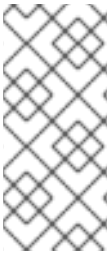
When specifying the **runtime-name** attribute with the **--runtime-name** option, you must include the **.war** extension in the name or the web context will not be registered by JBoss EAP. For example:

```
--runtime-name=my-application.war
```

3.5. SECURITY

JAAS realm in the elytron subsystem

In JBoss EAP 8.0, the legacy security subsystem has been removed. To continue using your custom login modules with the **elytron** subsystem, use the new Java Authentication and Authorization Service (JAAS) security realm, **jaas-realm**.



NOTE

jaas-realm only supports JAAS-compatible login modules. For information about JAAS, see Java Authentication and Authorization Service (JAAS) Reference Guide .

jaas-realm does not support custom login modules that extend or are dependent upon PicketBox APIs.

Although **elytron** subsystem provides **jaas-realm**, it is preferable to use other existing security realms that the subsystem provides. These include **jdbc-realm**, **ldap-realm**, **token-realm**, and others. You can also combine different security realms by configuring **aggregate-realm**, **distributed-realm**, or **failover-realm**. If none of these suits your purpose, implement a custom security realm and use it instead of custom login module.

The following are cases where you should use **jaas-realm** instead of implementing a custom security realm:

- You are migrating to the **elytron** subsystem from legacy security and already have custom login modules implemented.
- You are migrating from other application servers to JBoss EAP and already have the login modules implemented.
- You require combining multiple login modules with various flags and options provided to those login modules. These flags and options might not be configurable for the provided security realms in the **elytron** subsystem.

For more information, see Creating a JAAS realm in the *Securing applications and management interfaces using multiple identity stores* guide.

Configure multiple certificate revocation lists in Elytron and Elytron client

You can now configure multiple certificate revocation lists (CRL) in the **elytron** subsystem and WildFly Elytron client when you use several Certificate Authorities (CA). You can specify the list of CRLs to use in the **certificate-revocation-lists** attribute in the **trust-manager**.

For more information, see Configuring certificate revocation checks in Elytron in the *Configuring SSL/TLS in JBoss EAP* guide.

Keycloak SAML adapter feature pack

The archive distribution of Keycloak SAML adapter is no longer provided with JBoss EAP. Instead, you can use the Keycloak SAML adapter feature pack to install the **keycloak-saml** subsystem and related configurations.

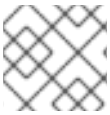
The Keycloak SAML adapter feature pack provides the following layers that you can install depending on your use case:

- **keycloak-saml**
- **keycloak-client-saml**
- **keycloak-client-saml-ejb**

For more information, see *Using single sign-on with JBoss EAP* guide.

Native OpenID Connect client

JBoss EAP now provides native support for OpenID Connect (OIDC) with the **elytron-oidc-client** subsystem. Therefore, Red Hat build of Keycloak Client Adapter is not provided in this release. The **elytron-oidc-client** subsystem acts as the Relying Party (RP). The **elytron-oidc-client** subsystem supports bearer-only authentication, and also provides multi-tenancy support. You can use the multi-tenancy support, for example, to authenticate users for an application from multiple Red Hat build of Keycloak realms.



NOTE

The JBoss EAP native OIDC client does not support RP-Initiated logout.

You can use the **elytron-oidc-client** subsystem to secure applications deployed to JBoss EAP and the JBoss EAP management console with OIDC.

Additionally, you can propagate the security identity, obtained from an OIDC provider, from a Servlet to Jakarta Enterprise Beans in both of the following cases:

- The Servlet and the Jakarta Enterprise Beans are in the same deployment.
- The Servlet and the Jakarta Enterprise Beans are in different deployments.

For more information, see *Using single sign-on with JBoss EAP* guide.

New hash-encoding and hash-charset attributes for hashed passwords

You can now specify the character set and the string format for the hashed passwords that are stored in **elytron** subsystem security realms by using the **hash-charset** and **hash-encoding** attributes. The default **hash-charset** value is **UTF-8**. You can set the **hash-encoding** value to either **base64** or **hex**; **base64** is the default for all realms except the **properties-realm** where **hex** is the default.

The new attributes are included in the following security realms:

- **filesystem-realm**
- **jdbc-realm**
- **ldap-realm**
- **properties-realm**

For more information, see the *Securing applications and management interfaces using an identity store* guide.

New encoding attribute for Elytron file-based audit log

You can now specify the encoding for file-based audit logs in Elytron by using the **encoding** attribute. The default value is **UTF-8**. The following values are possible:

- **UTF-8**
- **UTF-16BE**
- **UTF-16LE**
- **UTF-16**
- **US-ASCII**
- **ISO-8859-1**

For more information, see Elytron audit logging in the *Securing applications and management interfaces using an identity store* guide.

SSLv2Hello

Beginning with JBoss EAP 8.0 Beta, you can specify the **SSLv2Hello** protocol for **server-ssl-context** and **client-ssl-context** in the **elytron** subsystem.

WARNING

- You must configure another encryption protocol if you want to configure **SSLv2Hello** because the purpose of the latter is to determine which encryption protocols the connected server supports.
- **IBM JDK** does not support **SSLv2Hello** in its client, although a server-side connection always accepts this protocol.

Updates to filesystem-realm

You can now encrypt the clear passwords, hashed passwords, and attributes associated with identities in a **filesystem-realm** for better security. You can do this in two ways:

- Create an encrypted **filesystem-realm** by referencing a secret key in the **add** operation.
- Encrypt an existing **filesystem-realm** using the new **filesystem-realm-encrypt** command in the WildFly Elytron Tool.

You can now also enable integrity checks for a **filesystem-realm** to ensure that the identities in the **filesystem-realm** were not tampered with since the last authorized write. You can do this by referencing a key pair when you create the **filesystem-realm** using the **add** operation. WildFly Elytron generates a signature for the identity file using the key pair. An integrity check runs whenever an identity file is read.

For more information, see Filesystem realm in Elytron in the *Securing applications and management interfaces using an identity store* guide.

Updates to distributed-realm

You can now configure **distributed-realm** to continue searching the referenced security realms even when the connection to any identity store fails by setting the new attribute **ignore-unavailable-realms** to **true**.

By default, in case the connection to any identity store fails before an identity is matched, the authentication fails with an exception **RealmUnavailableException** as before.

When you set **ignore-unavailable-realms** to **true**, a **SecurityEvent** is emitted in case any of the queried realms are unavailable. You can configure this behavior by setting **emit-events** to **false**.

For more information, see the following resources in the *Securing applications and management interfaces using multiple identity stores* guide:

- Distributed realm in Elytron
- **distributed-realm** attibutes

Elytron support provided for SSLContexts in Artemis

In JBoss EAP 8, Elytron support is provided to instantiate the **SSLContext** variable in Messaging subsystem. This feature saves you from configuring **SSLContext** in multiple places as Elytron instantiates this variable. The connectors for the SSLContext must be defined on the **elytron** subsystem of the client's JBoss EAP server, which means that you cannot define it from a standalone messaging client application.

New Elytron client java security provider

Elytron client now provides a Java security provider, **org.wildfly.security.auth.client.WildFlyElytronClientDefaultSSLContextProvider**, that you can use to register a Java virtual machine (JVM)-wide default **SSLContext**.

When you register the provider in your JVM with high enough priority, then all client libraries that use **SSLContext.getDefault()** method obtain an instance of the SSL context that is configured to be default in Elytron client configuration. This way you can make use of Elytron client's SSL context configuration without interacting with Elytron API directly.

For more information, see Using Elytron client default SSLcontext security provider in JBoss EAP clients in the *Configuring SSL/TLS in JBoss EAP* guide.

Ability to obtain custom principal from Elytron

In JBoss EAP 8.0, you can now obtain a custom principal from Elytron. Previously, Elytron required principal to be an instance of **NamePrincipal** for authentication. While it was possible to use **SecurityIdentity** obtained from the current **SecurityDomain** and utilize **SecurityIdentity** attributes to obtain information from realms, it required reliance on **SecurityDomain** and **SecurityIdentity** instead of more generic and standardized methods like **jakarta.security.enterprise.SecurityContext.getCallerPrincipal()**.

You can now obtain a custom principal from the **getCallerPrincipal()** method when using Elytron. If your application code using legacy security relies on getting a custom principal from the **getCallerPrincipal()** method, you can migrate your application without requiring code changes.

3.6. CLUSTERING

Configuring web session replication using a ProtoStream

You can now configure web session replication using a ProtoStream instead of JBoss Marshalling in JBoss EAP 8.0.

See How to configure web session replication to use ProtoStream instead of JBoss Marshalling in JBoss EAP 8.0.

Stopping batch job execution from a different node

You can now stop batch job execution from a different clustered node in JBoss EAP 8.0. For more information see using Batch Processing **JBeret** with a clustering of nodes sharing the same job repository in JBoss EAP 8.0.

3.7. JAKARTA EE

Jakarta EE Core Profile

Jakarta EE 10 Core Profile is now available in JBoss EAP 8.0. The Core Profile is a small, lightweight profile that provides Jakarta EE specifications suitable for smaller runtimes, such as microservices and cloud services. The Jakarta EE 10 Core Profile is available as a Galleon provisioning layer, **ee-core-profile-server**.

For more information about the Core Profile Galleon layer, see Capability trimming in JBoss EAP for OpenShift: Base layers.

3.8. DATASOURCE SUBSYSTEM

Configuring custom exception-sorter or valid-connection-checker for a datasource

You can now configure a custom **exception-sorter** or **valid-connection-checker** for a datasource using a JBoss Module.

See How to configure a custom exception-sorter or valid-connection-checker for a datasource in JBoss EAP 8.

Support for eap-datasources-galleon-pack for JBoss EAP 8.0

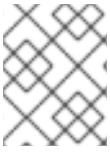
You can now use the **eap-datasources-galleon-pack** Galleon feature-pack to provision a JBoss EAP 8.0 server that can connect to your databases.

3.9. HIBERNATE

Hibernate Search 6 replaces Hibernate Search 5 APIs

Hibernate Search 5 APIs have been removed and are replaced with Hibernate Search 6 APIs in JBoss EAP 8.0.

To view a list of the removed features, see Hibernate Search 5 APIs Deprecated in JBoss EAP 7.4 and removed in EAP 8.0.



NOTE

Hibernate Search 6 APIs are **backwards-incompatible** with Hibernate Search 5 APIs. You will need to migrate your applications to Hibernate Search 6.

The latest version of Hibernate Search 6 included in JBoss EAP 8.0 is 6.2. If you are migrating from Hibernate Search 5, you should take into account the migration to version 6.0, 6.1, **and** 6.2.

See the following migrations guides for more information:

- To migrate your applications from Hibernate Search 5, see the Hibernate Search 6.0 migration guide.

- To migrate your applications from Hibernate Search 6.0 to 6.1, see the Hibernate Search 6.1 migration guide.
- To migrate your applications from Hibernate Search 6.1 to 6.2, see the Hibernate Search 6.2 migration guide

NOTE

Hibernate Search 6.2 is compatible with Hibernate ORM 6.2. For more information, see the section Hibernate ORM 6 in the Hibernate Search 6.2 Reference documentation.

Hibernate Search 6 supports Elasticsearch

JBoss EAP 8.0 also provides support for using an Elasticsearch backend in Hibernate Search 6 to index data into remote Elasticsearch or OpenSearch clusters.

To see a list of possible Hibernate Search architectures and backends, see Table 2. Comparison of architectures in the Hibernate Search 6.2 reference documentation.

For more information about configuring Hibernate Search 6, see Using Hibernate Search in the WildFly Developer guide.

3.10. INFINISPAN

Support for Infinispan distributed query, counter, and lock APIs and CDI modules

You can now use the Infinispan APIs for distributed query, counters, and locks in JBoss EAP 8.0.

The Infinispan CDI module is also available in JBoss EAP 8.0 for creating and injecting caches.

For more information, see EAP 8 now supports Infinispan query, counters, locks, and CDI .

3.11. MESSAGING

Addition of a new Galleon layer

A new Galleon layer is added to provide support for the Jakarta Messaging Service (JMS) integration with an embedded ActiveMQ Artemis broker. For more information, refer to the section Galleon layer for embedded broker messaging in the Migration Guide.

3.12. WEB SERVER (UNDERTOW)

Configuring a cookie for web request affinity

You can now configure a separate cookie to store session affinity information for load balancers by using the **affinity-cookie** resource at the address **/subsystem=undertow/servlet-container=default/setting=affinity-cookie**.

For more information, see the Red Hat Knowledgebase solution How to configure the affinity-cookie and session-cookie in JBoss EAP 8.

3.13. EJB3 SUBSYSTEM

JBoss EAP 8.0 server interoperability with JBoss EAP 7 and JBoss EAP 6

In JBoss EAP 8.0 you can enable interoperability between JBoss EAP 8.0 and older versions of your JBoss EAP server. JBoss EAP supports Jakarta EE 10 whose API class uses the **jakarta** package namespace. However, older versions of JBoss EAP use the **javax** package namespace.

IMPORTANT

- The older versions supported are JBoss EAP 6 and JBoss EAP 7
- interoperability between JBoss EAP 6 and JBoss EAP 7 is not affected by this issue as both servers support the **javax** package namespace.

For more information about how to enable interoperability between JBoss EAP 8.0 and older versions of JBoss EAP see, how to enable interoperability .

Infinispan-based distributed timers

In JBoss EAP 8.0, you can now use Infinispan-based distributed timers to schedule persistent Jakarta Enterprise Bean timers within a cluster, which you can scale to large clusters. For more information, see EAP 8 - how to configure Infinispan based distributed timers .

Distributable EJB subsystem

Use the **distributable-ejb** subsystem to configure clustering abstractions providers required for **ejb3** subsystem functionalities, such as:

- Stateful session beans (SFSB) cache factories
- Client mappings registries for EJB client applications
- Distributed EJB timers

You can currently define these providers at a system-wide level. It is planned to develop functionality to enable deployment-specific providers by customizing the **ejb3** subsystem. For more information, see What is the distributable-ejb subsystem in EAP 8 .

3.14. OPENSHIFT

Red Hat build of Keycloak SAML support for JBoss EAP 8.0

Using Red Hat build of Keycloak SAML adapters with JBoss EAP 8.0 Source-to-Image (S2I) image will be supported when the adapters are released. For more information, see OpenShift, SSO SAML support for EAP 8.

Provisioning a JBoss EAP server using the Maven plug-in

You can now use the JBoss EAP Maven plug-in on OpenShift to:

- Provision a trimmed server using Galleon.
- Install your application on the provisioned server.
- Tune the server configuration using the JBoss EAP management CLI.
- Package extra files into the server installation, such as a **keystore** file.
- Integrate the plug-in into your JBoss EAP 8.0 source-to-image application build.

For more information, see Provisioning a JBoss EAP server using the Maven plug-in .

OpenID Connect support for JBoss EAP source-to-image

You can now secure applications deployed to JBoss EAP with OpenID Connect (OIDC) using the new **elytron-oidc-client** subsystem instead of installing the previously required Red Hat build of Keycloak Client Adapter. You can configure an **elytron-oidc-client** subsystem by using the environment variables

to secure the application with OIDC. The Red Hat build of Keycloak Client Adapter is not provided in this release. For more information, see Using OpenID Connect to secure JBoss EAP applications on OpenShift.

Building application images using Source-to-Image

In JBoss EAP 8.0, an installed server has been removed from Source-to-Image (S2I) builder images. Galleon feature-packs and layers are now used to provision the server during the S2I build phase. To provision the server, include and configure the JBoss EAP Maven plug-in in the **pom.xml** file of your application. For more information, see Building application images using source-to-image in OpenShift .

Override management attributes with environment variables

To more easily adapt your JBoss EAP server configuration to your server environment, you can use an environment variable to override the value of any management attribute, without editing your configuration file. You cannot override management attributes of type **LIST**, **OBJECT**, or **PROPERTY**. In JBoss EAP 8.0 OpenShift runtime image, this feature is enabled by default. For more information, see Overriding management attributes with environment variables.

Environment variable checks for resolving management model expressions

JBoss EAP now supports environment variable checks when resolving management model expressions. In previous versions of JBoss EAP, the JBoss EAP server only checked for Java system properties in the management expression. Now, the server checks for relevant environment variables and system properties. If you use both, JBoss EAP will use the Java system property, rather than the environment variable, to resolve the management model expression. For more information about using environment variables to resolve management model expressions, see Using environment variables and model expression resolution.

Maven compatibility

Maven, versions 3.8.5 or earlier, include a version of the Apache Maven WAR plugin that is earlier than 3.3.2. This causes packaging errors with **eap-maven-plugin**. To resolve this issue, you must upgrade to Maven version 3.8.6 or later. Alternatively, you can add the **maven-war-plugin** dependency, version 3.3.2 or later, to your application **pom.xml**.

Enhancements to node naming

The value of the **jboss.node.name** system property is generated from the pod hostname and can be customized by using the **JBOSS_NODE_NAME** environment variable. This system property does not serve anymore as a transaction ID and does not have a limit of 23 characters in length, as it used to be in previous versions of JBoss EAP.

However, in JBoss EAP 8.0, a new system property, **jboss.tx.node.id**, is also generated from the pod hostname and can be customized by using the **JBOSS_NODE_NAME** environment variable. This system property is now limited to 23 characters in length and serves as the transaction ID.

Changes to Java options in JBoss EAP 8.0 images

The JVM automatically tunes the memory and cpu limits and Garbage Collector configuration in JBoss EAP 8.0 images. Instead of computing **-Xms** and **-Xmx** options, images use **-XX:InitialRAMPercentage** and **-XX:MaxRAMPercentage** options to achieve the same capability dynamically. **CONTAINER_CORE_LIMIT** and **JAVA_CORE_LIMIT** have been removed. Additionally, **-XX:ParallelGCThreads, -Djava.util.concurrent.ForkJoinPool.common.parallelism**, and **-XX:CICompilerCount** are no longer used.

Deploying a third-party application on OpenShift

With JBoss EAP 8.0, you can create application images for OpenShift deployments by using compiled WAR files or EAR archives. By using a Dockerfile, you can deploy these archives to a JBoss EAP server with the complete runtime stack, including the operating system, Java, and JBoss EAP components. You can create the application image without depending on Source-to-Image (S2I).

Excluded files in the JBoss EAP 8.0 server installation on OpenShift

When installing JBoss EAP 8.0 server on OpenShift, the following files are not required and are intentionally excluded:

- **bin/appclient.sh**
- **bin/wsprovide.sh**
- **bin/wsconsume.sh**
- **bin/jconsole.sh**
- **bin/client**

3.15. OPERATOR

Enhanced Health Probe configuration with JBoss EAP 8.0 Operator

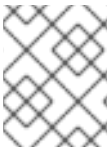
The JBoss EAP 8.0 Operator now offers improved configuration options for health probes, focusing on better probe customization and compatibility between JBoss EAP 8.0 and JBoss EAP 7.4 images. This enhancement ensures smooth interoperability between both images, allowing probes to adjust their execution method flexibly.

Key Improvements in your JBoss EAP 8.0 instance:

- Ability to work with JBoss EAP 8.0 and JBoss EAP 7-based images.
- Ability to configure **LivenessProbe**, **ReadinessProbe**, and **StartupProbes**.

Startup Probe example configuration:

```
apiVersion: wildfly.org/v1alpha1
kind: WildFlyServer
metadata:
  name: ...
spec:
  applicationImage: '...'
  livenessProbe:
    httpGet:
      path: /health/live
      port: 9990
      scheme: HTTP
    initialDelaySeconds: 30
  readinessProbe:
    httpGet:
      path: /health/ready
      port: 9990
      scheme: HTTP
    initialDelaySeconds: 10
  replicas: 1
  startupProbe:
    httpGet:
      path: /health/started
      port: 9990
      scheme: HTTP
    initialDelaySeconds: 60
```



NOTE

By default, JBoss EAP 8.0 applications retain shell probes to ensure backward compatibility for JBoss EAP 7-based applications.

3.16. QUICKSTARTS AND BOMS

Supported EAP 8 quickstarts

All supported JBoss EAP 8 quickstarts are located at jboss-eap-quickstarts.

New JBoss EAP BOMs for Maven

JBoss EAP BOMs provide the Maven BOM files that specify the versions of JBoss EAP dependencies that are needed for building or testing your Maven projects. In addition, Jakarta EE 10 BOMs provide dependency management for related frameworks such as Hibernate, RESTasy, and proprietary components like Infinispan and Client BOMs.

3.17. SERVER MIGRATION TOOL

JBoss EAP Server Migration Tool

The Server Migration Tool is now a standalone migration tool and is no longer included with JBoss EAP 8.0. You can download the migration tool separately.

3.18. ACTIVEMQ ARTEMIS

Failure to add bridge on the ActiveMQ server

In JBoss EAP 7, you could create a Java Message Service (JMS) bridge in the **messaging-activemq** subsystem before creating the source queue. The bridge remained inactive until the source queue was created.

In JBoss EAP 8, you must create the source queue before creating a JMS bridge with the **bridge:add** command. If you create the JMS bridge before you create the source queue, the **bridge:add** command will fail.

Adding a new connector in the messaging-activemq subsystem

In JBoss EAP 8.0, when a new connector is added to a configuration model using the CLI in the **messaging-activemq** subsystem, you must restart or reload the server so that the connector can be accessed by the other parts of the system. In JBoss EAP 7.4, a connector would be added and referenced by other parts of the system but it cannot be used without restarting or reloading the server.

3.19. JAKARTA FACES IMPLEMENTATION

Changes in Jakarta Faces implementation for MyFaces

In previous releases, you could replace the Jakarta Faces implementation with an alternative. However, for **MyFaces** in JBoss EAP 8.0, this functionality has been moved to an external feature pack that requires provisioning by using the Galleon tool. If you want to use a non-default Mojarra version, manual configuration is necessary. For more information, see How to configure the Multi-JSF feature in EAP 8 .

3.20. HIGH AVAILABILITY

Updates to the JGroup protocol stack

A new "RED" protocol has been added to the JGroup protocol stack in JBoss EAP 8.0. Additionally, the existing protocols have been upgraded.

The following table lists the protocol updates:

Old protocol	Upgraded protocol
FD_SOCK	**FD_SOCK2**
FD_ALL	**FD_ALL3**
VERIFY_SUSPECT	**VERIFY_SUSPECT2**
FRAG3	**FRAG4**

While the old protocol stack will still work in JBoss EAP 8.0, use the upgraded stack for optimal results.

3.21. THE JBOSS-EAP-INSTALLATION-MANAGER

You can now install and update JBoss EAP 8.0 using the **jboss-eap-installation-manager**. You can also perform server management operations, including updating, reverting, and various channel management tasks.

For more information, see The Installation guide.

3.22. MANAGEMENT CLI INTEGRATION OF JBOSS-EAP-INSTALLATION-MANAGER

In JBoss EAP 8.0, a significant enhancement has been introduced with the integration of the **jboss-eap-installation-manger** with the Management CLI under the **installer** command. This enhancement allows you to seamlessly perform a wide range of server management operations such as updating, reverting, and managing channel operations in a standalone or a managed domain mode.

For more information, see The Update guide.

3.23. WEB CONSOLE INTEGRATION OF JBOSS-EAP-INSTALLATION-MANAGER

In JBoss EAP 8.0, you can now use the web console to update, revert, and manage channels in your JBoss EAP installation. However, it is recommended to use the **jboss-eap-installation-manager**.

For more information, see The Update guide.

3.24. JBOSS EAP APPLICATION MIGRATION

If you have used the **galleon/provisioning.xml** configuration file, to provision your JBoss EAP 7.4 installation with a valid S2I and you want to convert the file to a valid configuration for JBoss EAP 8 you must take note of the following changes:

- In your **galleon/provisioning.xml** configuration file you must use the **org.jboss.eap:wildfly-ee-galleon-pack** and **org.jboss.eap:eap-cloud-galleon-pack** feature packs instead of the **eap-s2i** feature pack.

- To successfully use these feature packs, you must also enable the use of JBoss EAP 8 channels by either configuring the **eap-maven-plugin** in the application **pom.xml** or using the S2I environment variable.

Additional resources

- The Galleon provisioning file.
- Creating an S2I build using the legacy S2I provisioning capabilities .
- The Maven plug-in configuration attributes .

CHAPTER 4. UNSUPPORTED, DEPRECATED, AND REMOVED FUNCTIONALITY

4.1. UNSUPPORTED FEATURES

The following features are not supported by Red Hat.

Logging

JBoss EAP 8.0 does not support Apache Log4j version 1 APIs. If your applications do not package **log4j.jar** and Log4j configuration as part of the application, then you must update them. For more information about migrating or updating your applications, see the Red Hat Knowledgebase solution Migration: Apache Log4j version 1 is no longer provided in EAP 8 .

Agroal subsystem

JBoss EAP 8.0 no longer supports the Agroal subsystem.

4.2. DEPRECATED FEATURES

Some features are deprecated with this release. This means that no enhancements will be made to these features, and they might be removed in a future release. For more information, see Deprecated in Red Hat JBoss Enterprise Application Platform (EAP) 8.

Red Hat will continue providing full support and bug fixes under our standard support terms and conditions. For more information about the Red Hat support policy, see the Red Hat JBoss Middleware Product Update and Support Policy located on the Red Hat Customer Portal.

The following features are deprecated:

JBoss Tools

JBoss Tools is deprecated in JBoss EAP 8.0.

4.3. REMOVED FEATURES

JBoss EAP 8.0 removes the following features.

Jolokia and Prometheus

Jolokia and **Prometheus** have been removed in this release. These features have been dropped and will no longer be supported by Red Hat. JBoss EAP server exposes metrics through the server metrics endpoint: **<server address>:<management port>/metrics**.

Environment variables

Red Hat has removed the following environment variables in JBoss EAP 8.0:

- **GALLEON_PROVISION_DEFAULT_FAT_SERVER**
- **AB_JOLOKIA_AUTH_OPENSHIFT**
- **AB_JOLOKIA_CONFIG**
- **AB_JOLOKIA_DISCOVERY_ENABLED**
- **AB_JOLOKIA_HOST**
- **AB_JOLOKIA_HTTPS**

- **AB_JOLOKIA_ID**
- **AB_JOLOKIA_OFF**
- **AB_JOLOKIA_OPTS**
- **AB_JOLOKIA_PASSWORD**
- **AB_JOLOKIA_PASSWORD_RANDOM**
- **AB_JOLOKIA_PORT**
- **AB_JOLOKIA_USER**
- **AB_PROMETHEUS_ENABLE**
- **AB_PROMETHEUS_JMX_EXPORTER_CONFIG**
- **AB_PROMETHEUS_JMX_EXPORTER_PORT**
- **JGROUPS_ENCRYPT_SECRET**

JDK 8

JDK 8 has been removed from Red Hat JBoss Enterprise Application Platform 8.0. JDK 11 or JDK 17 is now required.

Legacy security realms

The legacy security realms have been removed from JBoss EAP 8.0. Use the security realms provided in the **elytron** subsystem instead.

For more information, see the *Securing applications and management interfaces using an identity store* , and *Securing applications and management interfaces using multiple identity stores* guides.

Picketbox

PicketBox has been removed from Red Hat JBoss Enterprise Application Platform 8.0. Any legacy security configurations must be migrated to the **elytron** subsystem. For more information about migrating your security configurations to the **elytron** subsystem, see Migrating to Elytron.

PicketBox vault

PicketBox vault has been removed from JBoss EAP 8.0. Use the credential store provided by the **elytron** subsystem to store sensitive strings instead.

For more information, see Credentials and credential stores in Elytron in the *Secure storage of credentials in JBoss EAP* guide.

PicketLink Subsystem

The PicketLink subsystem has been removed from JBoss EAP 8.0. Use Red Hat build of Keycloak instead of the PicketLink identity provider, and the Galleon layers provided by the Keycloak SAML adapter feature instead of the PicketLink service provider.

For more information, see Securing applications with SAML in the *Using single sign-on with JBoss EAP* guide.

discovery-group and **broadcast-group** resources

Red Hat JBoss Enterprise Application Platform 7.4 removed the **discovery-group** and **broadcast-group** resources. These resources are still removed in JBoss EAP8.0.

Additionally, Red Hat JBoss Enterprise Application Platform 7.4 reduced the impact to its web console by replacing all instances of **discovery-group** and **broadcast-group** resources with **jgroups-discovery-group** and **socket-discovery-group** resources.

JBoss EAP 7.3 deprecated the following resources in the **messaging** subsystem:

- **/subsystem=messaging-activemq/discovery-group=***
- **/subsystem=messaging-activemq/server=default/broadcast-group=***
- **/subsystem=messaging-activemq/server=default/discovery-group=***

JBoss EAP 7.3 replaced these deprecated resources with **jgroups-discovery-group** and **socket-discovery-group** resources. Each deprecated resource included an attribute from each replacement resource, with one attribute set to **null** and the other attribute set to a value greater than **0**. These settings caused both **discovery-group** and **broadcast-group** to remain active, but still assign all their functionality to the **jgroups-discovery-group** and **socket-discovery-group** resources.

Quickstarts

The following outdated or redundant quickstarts have been removed from JBoss EAP 8.0:

- **app-client**
- **bean-validation**
- **ejb-asynchronous**
- **ejb-in-ear**
- **ejb-in-war**
- **ejb-security**
- **ejb-security-jaas**
- **greeter**
- **helloworld-html5**
- **helloworld-mbean**
- **helloworld-mdb-propertysubstitution**
- **helloworld-rs**
- **helloworld-ssl**
- **inter-app**
- **jaxws-addressing**
- **jaxws-pojo**
- **jts-distributed-crash-rec**
- **kitchensink-angularjs**
- **kitchensink-ear**

- **kitchensink-jsp**
- **kitchensink-ml**
- **logging-tools**
- **managed-executor-service**
- **messaging-clustering**
- **payment-cdi-event**
- **resteasy-jaxrs-client**
- **spring-greeter**
- **spring-kitchensink-basic**
- **spring-kitchensink-springmvctest**
- **tasks-rs**
- **websocket-client**
- **xml-jaxp**

Red Hat build of Keycloak Client Adapter

Red Hat JBoss Enterprise Application Platform 8.0 does not provide the Red Hat build of Keycloak Client Adapter. Use the new **elytron-oidc-client** subsystem to secure applications deployed to JBoss EAP with OpenID Connect (OIDC).

Java service on Red Hat Enterprise Linux

Java service (JSVC) running on Red Hat Enterprise Linux (RHEL) has been removed from JBoss EAP 8.0.

BOMs

The following BOMs have been removed:

- The **JBoss Jakarta EE 8 Specification APIs** BOM is removed. Use the **JBoss EAP EE** BOM in your Maven project.
- The **EAP Runtime Artifacts** BOM is removed. Use the **JBoss EAP EE** BOM in your Maven project.
- The **JBoss EJB client legacy** BOM is removed.

For more information, see Migrate a JBoss EAP Application's Maven Project to JBoss EAP 8.0 in the *Migration Guide*.

Connector attribute

In JBoss EAP 7.4, the modcluster subsystem deprecates the **connector** attribute on the proxy element and replaces it with the **listener** attribute to avoid confusion. The management schema in JBoss EAP 7.4 uses the **listener** attribute, but also allows setting the **connector** attribute. In JBoss EAP 8.0, the deprecated **connector** attribute is removed, and you must now use the **listener** attribute instead. For more information, see Deprecated in Red Hat JBoss Enterprise Application Platform (EAP) 7 .

Changes to the iiop-openjdk subsystem

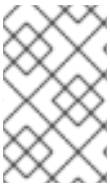
In JBoss EAP 8.0, the legacy security subsystem has been removed and replaced by the **elytron** subsystem. You can install the **elytron** subsystem as the security interceptor for Object resource broker (ORB).

To maintain interoperability with JBoss EAP 7 , for example when running JBoss EAP as a managed domain with a JBoss EAP 8.0 host controller managing a JBoss EAP 7 secondary host controller, the ability to configure legacy security interceptors has been retained.

However, when running JBoss EAP as a standalone server, setting the values **client** and **identity**, for **iiop-openjdk/security** attribute is **not** supported.

Hibernate Search 5 APIs

Hibernate Search 5 APIs were deprecated in JBoss EAP 7.4 and have been removed in JBoss EAP 8.0 and replaced with Hibernate Search 6 APIs.



NOTE

Hibernate Search 6 APIs are **backwards-incompatible** with Hibernate Search 5 APIs. You will need to migrate your applications to Hibernate Search 6 to maintain operability. To migrate your applications, see the Hibernate Search 6.0 migration guide.

To view a list of the removed APIs, see Hibernate Search 5 APIs Deprecated in JBoss EAP 7.4 and removed in EAP 8.0.

Apache Log4j version 1

JBoss EAP 8.0 does not support Apache Log4j version 1 APIs. If your applications do not package **log4j.jar** and Log4j configuration as part of the application, then you must update these packages. For more information about migrating or updating your applications, see the Red Hat Knowledgebase solution Migration: Apache Log4j version 1 is no longer provided in EAP 8 .

For more information, see Removal of Apache Log4j version 1 APIs .

Apache Xerces and Apache Xalan

The Apache Xerces and Apache Xalan JBoss Modules, implementing JAXP version 1.5, have been removed from JBoss EAP 8.0. Use the default JAXP implementation provided by the **java.xml** JPMS module of JDK instead, which implements JAXP version 1.6. For more information, see Use JAXP implementation provided by the JDK in JBoss EAP.

CHAPTER 5. RESOLVED ISSUES

See Resolved Issues for Red Hat JBoss Enterprise Application Platform 8.0 to view the list of issues originating from customer cases that have been resolved for this release.

CHAPTER 6. FIXED CVES

JBoss EAP 8.0 includes fixes for the following security-related issues:

- JBEAP-25077:(8.0.z) Upgrade netty from 4.1.92.Final to 4.1.94.Final (resolves CVE-2023-34462)
- JBEAP-25748:(8.0.z) JBWS-4389 - Wrong assumption about the Identity's password are all clearpassword
- JBEAP-25356:(8.0.0) Upgrade Guava from 31.1.0.jre-redhat-00001 to 32.1.1.jre-redhat-00001
- JBEAP-25646:(8.0.z) Upgrade FasterXML Jackson from 2.14.2.redhat-00001 to 2.15.2
- JBEAP-25943: Upgrade santuario to 3.0.3
- JBEAP-24435: MP OpenAPI - Loading static files bigger than 3MB fails since SmallRye OpenAPI 3.0.1 uses new SnakeYaml that sets a constraint

CHAPTER 7. KNOWN ISSUES

See Known Issues for Red Hat JBoss Enterprise Application Platform 8.0 to view the list of known issues for this release.

7.1. INFINISPAN

The /subsystem=distributable-web/infinispan-session-management=*:add operation may fail when executed on a default non-HA server configuration

Issue - JBEAP-24997

The **/subsystem=distributable-web/infinispan-session-management=*:add** operation automatically adds the **affinity=primary-owner** child resource, which requires the **routing=infinispan** resource. The operation may fail because the required **routing=infinispan** resource is not defined in the default non-HA server configurations.

Workaround

To avoid this invalid intermediate state, execute both **infinispan-session-management:add** and **affinity=local:add** operations within a batch.

Example:

```
batch
/subsystem=distributable-web/infinispan-session-management=ism-0:add(cache-container=web,granularity=SESSION)
/subsystem=distributable-web/infinispan-session-management=ism-0/affinity=local:add()
run-batch -v
```

HotRod cannot create distributed sessions for externalization to Infinispan

Issue - JBEAP-26062

An interoperability test involving Red Hat JBoss Enterprise Application Platform 8.0 and Red Hat Data Grid on OpenShift Container Platform shows an issue where writes to an Infinispan remote cache causes an internal server error. When the **remote-cache-container** is configured to use the default marshaller, JBoss Marshalling, cache writes cause HotRod to throw errors because only byte[] instances are supported.

Example error message:

```
Caused by: java.lang.IllegalArgumentException: Only byte[] instances are supported currently!
 at org.infinispan.client.hotrod@14.0.17.Final-redhat-00002//org.infinispan.client.hotrod.marshall.BytesOnlyMarshaller.checkByteArray(BytesOnlyMarshaller.java:27)
```

Workaround

Configure the **remote-cache-container** to use the ProtoStream marshaller **marshaller=PROTOSTREAM**:

Example configuration:

```
/subsystem=infinispan/remote-cache-container=
<RHDG_REMOTE_CACHE_CONTAINER_RESOURCE_NAME>:write-
attribute(name=marshaller,value=PROTOSTREAM)
```

7.2. DATASOURCE CONFIGURATION

MsSQL connection resiliency is not supported

Issue - JBEAP-25585

Red Hat JBoss Enterprise Application Platform 8.0 does not support connection resiliency of MsSQL JDBC driver version 10.2.0 and later. Connection resiliency causes the driver to be in an unexpected state for the recovery manager. By default, this driver has connection resiliency enabled and must be manually disabled by the user.

Workaround

The **ConnectRetryCount** parameter controls the number of reconnection attempts when there is a connection failure. This parameter is set to **1** by default, enabling connection resiliency.
To disable connection resiliency, change the **ConnectRetryCount** parameter from **1** to **0**. You can set connection properties in the datasource configuration section of the server configuration file **standalone.xml** or **domain.xml**. For more information about how to configure datasource settings, see How to configure datasource settings in EAP for OpenShift and How to specify connection properties in the Datasource Configuration for JBoss EAP on the Red Hat Customer Portal.

7.3. SERVER MANAGEMENT

Liveness probe :9990/health/live does not restart pod in case of Deployment Error

Issue - JBEAP-24257

In JBoss EAP 7.4, the python liveness probe reports "not alive" when there are deployment errors that would result in restarting the container.
In JBoss EAP 8.0, the liveness probe **:9990/health/live** uses the server management model to determine readiness. If the server-state is running, and there are no boot or deployment errors, then the liveness check reports **UP** when the server process is running.

Therefore, deployment errors can result in a pod that is running but is "not ready". This would only affect applications that have intermittent errors during deployment. If these errors always occur during deployment, the container will never be ready and the pod would be in a **CrashLoopBackoff** state.

NOTE

:9990/health/live is the default liveness probe used by Helm charts and the JBoss EAP operator.

Workaround

If there are deployment errors that result in a pod that is running but is reporting "not ready", examine the server boot process, resolve the deployment issue causing the errors, and then verify that the server deploys correctly.
If the deployment errors cannot be fixed, change the startup probe to use the **/ready** HTTP endpoint so that boot errors will trigger a pod restart. For example, if you deploy a JBoss EAP application with Helm, configure the liveness probe by updating the **deploy.livenessProbe** field:

```
deploy:
  livenessProbe:
    httpGet:
      path: /health/ready
```

7.4. MESSAGING FRAMEWORK

Deprecation of org.apache.activemq.artemis module and warning messages

Issue - JBEAP-26188

The **org.apache.activemq.artemis** module is deprecated in JBoss EAP 8.0. A warning message is triggered when deploying an application that include this module dependency in either the **MANIFEST.MF** or **jboss-deployment-structure.xml** configuration files. For more information, see Deprecated in Red Hat JBoss Enterprise Application Platform (EAP) 8 .

Workaround

With JBoss EAP 8.0 Update 1, you can prevent logging of these warning messages by replacing the **org.apache.activemq.artemis** module in your configuration files with the **org.apache.activemq.artemis.client** public module. For more information, see **org.jboss.as.dependency.deprecated** ... is using a deprecated module ("org.apache.activemq.artemis") in EAP 8.

7.5. IBM MQ RESOURCE ADAPTERS

Limitations and known issues of IBM MQ resource adapters

IBM MQ resource adapters are supported with some limitations. See Deploying the IBM MQ Resource Adapter for more information.

Revised on 2024-02-27 10:08:25 UTC