Abstract

These release notes contain important information related to Red Hat JBoss Enterprise Application Platform 7.4. Beginning with JBoss EAP 7.4, we will be exclusively focusing on the Jakarta EE test suite for certification/compliance. In order to support our existing customers who may be migrating their applications from the previous JBoss EAP 7 versions, JBoss EAP 7.4 maintains backwards compatibility. Oracle donated Java EE 8 TCKs to Jakarta EE which are used by Jakarta EE 8 implementations (like JBoss EAP) to become Jakarta EE 8-compatible. Jakarta EE 8 APIs are equivalent to Java EE 8 APIs. Jakarta EE 8 specifications (technologies) are equivalent to Java EE 8 specifications (technologies).
Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .......................................................... 5
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................. 6
CHAPTER 1. SUPPORTED CONFIGURATIONS ...................................................... 7
CHAPTER 2. NEW FEATURES AND ENHANCEMENTS ......................................... 8
  2.1. SECURITY ......................................................................................... 8
      Support for automatic update of credentials in a credential store ................. 8
      New role mapper regex-role-mapper in Elytron ........................................ 8
      Accessing IP address of remote client ..................................................... 8
      The aggregate-role-decoder role decoder ............................................... 8
      Using TLS protocol version 1.3 with JDK 11 ............................................ 9
      Enable support for the TLS 1.3 protocol with the OpenSSL provider for TLS 9
      Re-enable support for the TLS 1.1 protocol in your JDK configuration .......... 9
      Using SSH credentials to connect to a remote Git SSH repository ............. 9
      New principal transformer added to the elytron subsystem ...................... 9
      Ability to automatically generate a self-signed certificate .......................... 9
      Configuration of multiple security realms to support failover .................. 10
      Distributed identities across multiple security realms .............................. 10
      Access to external credentials over HTTP in the elytron subsystem .......... 10
      Use the Elytron client authentication configuration with the RESTEasy client 10
      Secret key credential store for providing initial secret key ....................... 10
      Encrypted expressions for securing security-sensitive strings .................. 10
      Updates to elytron-tool ......................................................................... 11
  2.2. SERVER MANAGEMENT ...................................................................... 11
      Use a global directory to distribute shared libraries across deployments ........ 11
      Support for read-only server configuration directories ............................ 11
      Ability to pass JBoss Module parameters .............................................. 11
      Infinispan APIs ................................................................................. 11
      Configurable option to allow requests during startup .............................. 12
      Configurable common script file added ............................................... 12
  2.3. MANAGEMENT CLI ............................................................................ 12
      Enhancement to the command CLI command .......................................... 12
      New role decoder added to the elytron subsystem ................................... 12
      Exposing runtime statistics for managed executor services ..................... 13
      Terminating hung tasks ...................................................................... 13
      Using property replacement for permissions files .................................. 13
      Configuring RESTEasy parameters ..................................................... 13
      Configuring RESTEasy providers ....................................................... 14
  2.4. MANAGEMENT CONSOLE .................................................................. 14
      New role decoder added to the elytron subsystem ................................... 14
  2.5. LOGGING ......................................................................................... 15
      The Apache Log4j2 API ...................................................................... 15
  2.6. EJB3 SUBSYSTEM ............................................................................. 15
      Default global stateful session bean timeout value in the ejb3 subsystem .... 15
      Forcing Jakarta Enterprise Beans timer refresh in database-data-store .... 15
      Access to runtime information from Jakarta Enterprise Beans ............... 16
      Dynamic discovery of Jakarta Enterprise Beans over HTTP .................... 16
      Global configuration of compression for remote Jakarta Enterprise Beans calls 16
      New attribute for setting the principal propagation behavior in Elytron .... 16
  2.7. HIBERNATE ....................................................................................... 16
Configuring the wildfly.jpa.skipquerydetach persistence unit property

2.8. WEB SERVICES
Integrating Elytron with web services clients
Ability for RESTEasy 3.x to access all standard MicroProfile ConfigSources
Configuring SameSite cookie attribute
Configuring Eclipse MicroProfile REST client API in resteasy CDI modules

2.9. MESSAGING
Duplicate messages on the JMS core bridge
Ability to pause a topic
Ability to detect network isolation of broker
call-timeout attribute
Red Hat AMQ connection pools

2.10. SCRIPTS
New environment variable for starting your server

2.11. OPENSHIFT
Providing custom Galleon feature-pack support to your JBoss EAP S2I image
Read-only server configuration directory
Instructions to deploy JBoss EAP quickstarts on OpenShift
New Galleon layer for the Distributable Web subsystem

2.12. RED HAT CODEREADY WORKSPACES (CRW)
Red Hat CodeReady Workspaces supports JBoss EAP 7.4 development files

CHAPTER 3. UNSUPPORTED FUNCTIONALITY

3.1. UNSUPPORTED FEATURES
Platforms and features
Keystore defect with Java jdk8u292-b10
RESTEasy parameters
MicroProfile capabilities
Red Hat JBoss Operations Network
MS SQL Server 2017

3.2. DEPRECATED FEATURES
Platforms and features
Operating systems
Databases and database connectors
Lightweight Directory Access Protocol (LDAP) servers
Spring BOM
BOMs
Java Development Kits (JDKs)
JBoss EAP OpenShift templates
eap74-beta-starter-s2i.json and eap73-third-party-db-s2i.json templates
Legacy security subsystem
PicketLink
PicketBox
Managed domain support for previous versions of JBoss EAP
Server configuration files using namespaces from JBoss EAP 7.3 and earlier
JBoss EAP Server Side JavaScript support
Agroal subsystem
Codehaus Jackson
application-security-domain resources
Clustering subsystems
Salted Challenge Response Authentication Mechanism
Quickstarts
Hibernate ORM 5.1
CHAPTER 4. RESOLVED ISSUES .................................................. 27

CHAPTER 5. FIXED CVES ..................................................... 28

CHAPTER 6. KNOWN ISSUES .................................................. 29

6.1. CHANGED BEHAVIORS FOR JBOSS EAP 7.4 29
   Setting OPENSHIFT_DNS_PING_SERVICE_NAME to an empty value results in boot error 29
   Unpredictable web session expiration 29
   Memory leaks in distributed JSF applications when caching managed beans in a WebInjestionContainer 29
   Java.lang.NotNullerException error when using ibm-java-1.8 and Bouncy Castle 29
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.
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. Follow the steps in the procedure to learn about submitting feedback on Red Hat documentation.

Prerequisites

- Log in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in Multi-page HTML format.

Procedure

1. Click Feedback to see existing reader comments.

   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. In the prompt menu that displays near the text you selected, click Add Feedback.
   A text box opens in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.
   You have created a documentation issue.

5. To view the issue, click the issue tracker link in the feedback view.
MS SQL Server 2019 is supported in JBoss EAP 7.4.

The Red Hat JBoss Enterprise Application Platform (EAP) 7 Supported Configurations knowledgebase article on the Red Hat Customer Portal lists databases and database connectors that were tested as part of the JBoss EAP 7.4 release.
CHAPTER 2. NEW FEATURES AND ENHANCEMENTS

2.1. SECURITY

Support for automatic update of credentials in a credential store
Elytron now automates adding and updating a credential to a previously defined credential store when
you configure a credential reference that specifies both the store and clear-text attributes.

With this update, you do not need to add a credential to an existing credential store before you can
reference it from a credential-reference. The automated process reduces the number of steps you
need to perform for referencing new credentials in different subsystems.

New role mapper regex-role-mapper in Elytron
Elytron now provides a new role mapper, regex-role-mapper, to define a regular expression (regex)
based mapping of security roles.

You can use regex-role-mapper to translate a list of roles to simpler roles. For example:

- *-admin to admin
- *-user to user

With regex-role-mapper, you do not need to implement your own custom component to translate
security roles.

Accessing IP address of remote client
You can now add the source-address-role-decoder role decoder to the elytron subsystem. By
configuring this role decoder, you can gain additional information from a remote client when making
authorization decisions.

The source-address-role-decoder extracts the IP address of a remote client and checks that it
matches the IP address specified in the pattern attribute or the source-address attribute. If the IP
address of the remote client matches the IP address specified in either attribute, the roles attribute
then assigns roles to the user. When you have configured source-address-role-decoder, you can
reference it in the role-decoder attribute of the security domain.

The aggregate-role-decoder role decoder
The aggregate-role-decoder consists of two or more role decoders. After each specified role decoder
completes its operation, it adds roles to the aggregate-role-decoder.

You can use aggregate-role-decoder to make authorization decisions by adding role decoders that
assign roles for a user. Further, aggregate-role-decoder provides you with a convenient way to
aggregate the roles returned from each role decoder.

Using TLS protocol version 1.3 with JDK 11
Elytron now provides the ability to use Transport Layer Security (TLS) Protocol version 1.3 for JBoss
EAP running against JDK 11.

TLS 1.3 is disabled by default. You can enable TLS 1.3 by configuring the new cipher-suite-names
attribute in the SSL Context resource definition in the elytron subsystem.

Compared with TLS 1.2, you might experience reduced performance when running TLS 1.3 with JDK 11.
Diminished performance might occur when a very large number of TLS 1.3 requests are being made. A
system upgrade to a newer JDK version can improve performance. Test your setup with TLS 1.3 for
performance degradation before enabling it in production.
Enable support for the TLS 1.3 protocol with the OpenSSL provider for TLS
JBoss EAP 7.4 includes support for the Transport Layer Security (TLS) protocol version 1.3. The use of TLS 1.3 protocol with the OpenSSL provider for TLS is disabled by default.

You can enable support for the TLS 1.3 protocol with the OpenSSL provider for TLS by configuring the `cipher-suite-names` attribute in the `ssl-context` configuration.

Compared with TLS 1.2, you might experience reduced performance when running TLS 1.3 with JDK 11. Diminished performance might occur when a very large number of TLS 1.3 requests are being made. A system upgrade to a newer JDK version can improve performance. Test your setup with TLS 1.3 for performance degradation before enabling it in production.

Re-enable support for the TLS 1.1 protocol in your JDK configuration
Newer versions of JDK might disable the Transport Layer Security (TLS) protocol version 1.1 by default. If your JBoss EAP 7.4 configuration must comply with the Federal Information Processing Standard (FIPS), you might need to re-enable support for the TLS 1.1 protocol in your JDK configuration.

For more information about TLS protocols compatible with JBoss EAP 7.4, see the Red Hat JBoss Enterprise Application Platform (EAP) 7 Supported Configurations page on the Red Hat Customer Portal.

Using SSH credentials to connect to a remote Git SSH repository
With JBoss EAP 7.4, you can use SSH credentials to connect to a remote Git SSH repository. This repository can manage your server configuration data, properties files, and deployments.

You must use the `elytron` configuration file to specify SSH credentials. You can then start your standalone server instance and have a remote Git SSH repository manage your server configuration file history.

If necessary, you can generate SSH keys by using one of the following methods:

- The `elytron-tool.sh` script
- The OpenSSH command line

For information about connecting to a remote Git SSH repository, see Using a remote Git SSH repository.

New principal transformer added to the `elytron` subsystem
JBoss EAP 7.4 includes a new principal transformer, `case-principal-transformer`, in the `elytron` subsystem. You can use the `case-principal-transformer` to change a principal’s username to either uppercase or lowercase characters.

Ability to automatically generate a self-signed certificate
With JBoss EAP 7.4, you can automatically generate a self-signed certificate.

Use a self-signed certificate only in a test environment. Do not use a self-signed certificate in a production environment.

To use this new feature, in the `undertow` subsystem, update the configuration of the `http-listener`.

```bash
batch
/subsystem=undertow/server=default-server/https-listener=https:undefine-attribute(name=security-realm)
/subsystem=undertow/server=default-server/https-listener=https:write-attribute(name=ssl-`
After you update the configuration, and if no keystore file exists, the first time JBoss EAP receives an HTTPS request, the system automatically generates a self-signed certificate. JBoss EAP logs a warning when a self-signed certificate is used.

Configuration of multiple security realms to support failover
With JBoss EAP 7.4, you can configure a failover security realm. If the security realm is not available, JBoss EAP uses the failover realm. The following code illustrates an example configuration:

```xml
<failover-realm name="myfailoverrealm" delegate-realm="LdapRealm" failover-realm="LocalRealm"/>
```

Distributed identities across multiple security realms
With JBoss EAP 7.4, you can configure a distributed security realm, which sequentially invokes a list of configured realms until a realm with the identity is found. The following code illustrates an example configuration:

```xml
<distributed-realm name="mymainrealm" realms="realm1 realm2 realm3"/>
```

Access to external credentials over HTTP in the elytron subsystem
With JBoss EAP 7.4, JBoss EAP can authenticate a user based on credentials established externally when using HTTP authentication.

To use this capability, configure a security domain to use the External mechanism when authenticating users.

Use the Elytron client authentication configuration with the RESTEasy client
The JBoss EAP 7.4 release integrates the RESTEasy client with the Elytron client. The RESTEasy client uses authentication information, such as credentials, bearer tokens, and SSL configurations, from an Elytron client configuration.

You can specify the Elytron client configuration that the RESTEasy client can use in the following ways:

- By providing the `wildfly-config.xml` file to the Elytron client. The Elytron client searches the class path for `wildfly-config.xml` or `META-INF/wildfly-config.xml`.
  - Alternatively, you can use the `wildfly.config.url` system property to specify the path for the `wildfly-config.xml` file.
- By using the Elytron client API to programmatically specify the authentication configuration.

Secret key credential store for providing initial secret key
You can now provide an initial secret key to the application server process using a new type of credential store named `secret-key-credential-store`. With this credential store, you get more robust security than password-based encryption because you can now manage your own initial secret.

Additionally, you can now generate secret keys, and also export and import previously generated secret keys, for all credential stores. You can also use existing credential stores for storing secret keys, and management operations to maintain them.

Encrypted expressions for securing security-sensitive strings
You can now use encrypted expressions to securely store security-sensitive strings in the management model. Elytron encrypts plain text strings using Advanced Encryption Standard (AES) encryption and
decrypts the encrypted expression dynamically at runtime using a `SecretKey` key stored in a credential store.

You can configure encrypted expressions using the new resource `expression-encryption` in the `elytron` subsystem. Use the `create-expression` management operation to create encrypted expressions.

**NOTE**

Use the credential store for storing passwords. The password vault is deprecated and will be removed in a future release.

Updates to `elytron-tool`

You can use the `elytron-tool` with both the existing and new credential stores. Use the `credential-store` command to manage secret keys and to generate encrypted tokens for use in expressions.

### 2.2. SERVER MANAGEMENT

**Use a global directory to distribute shared libraries across deployments**

In JBoss EAP 7.3 and earlier versions, you could not create and configure a global directory to distribute shared libraries across deployments running on a server. These capabilities have been added to the `ee` subsystem.

A global directory offers a better alternative to the global module approach. For example, if you want to change the name of a library listed in a global module, you must remove the global module, change the library's name, and then add the library to a new global module. If you change the name of a library that is listed in the global directory, you only need to restart the server to make the library name change available for all deployments.

Using a global directory is also a better solution if you want to share multiple libraries across deployed applications.

For more information, see Define global modules in the JBoss EAP Configuration Guide.

**Support for read-only server configuration directories**

In JBoss EAP 7.3 and earlier versions, servers fail to start if the configuration directory is configured as read-only. JBoss EAP 7.4 introduces the ability to use a read-only server configuration directory. If the configuration directory is read-only, include the `--read-only-server-config` switch in a command to start the server.

**Ability to pass JBoss Module parameters**

In the configuration files for JBoss EAP 7.3 and earlier versions, JBoss Modules did not include the ability to pass module parameters. In the script configuration files for JBoss EAP 7.4 you can now add a `MODULE_OPTS=-javaagent:my-agent.jar` environment variable to pass JBoss Module parameters.

You can use this capability when you previously were required to add the log manager on the boot class path.

**Infinispan APIs**

Previously, the Infinispan APIs were flagged as private within EAP as they are a part of the Red Hat Data Grid project. These APIs are now fully included and supported in JBoss EAP7.4. The modules included are:

- `org.infinispan`
- `org.infinispan.client.hotro`
• org.infinispan.commons

Configurable option to allow requests during startup
Added the option for a graceful startup mode for when user requests need to occur earlier in the startup process. This is supported for both managed domains, and standalone servers.

• For servers in a managed domain, the server-group element now supports the graceful-startup argument. The default for this is set to true.

• In a standalone server, set the command line option --graceful-startup=false to the required value.

Configurable common script file added
You can now use the file, common.conf, to customize your JBoss EAP instance environments. The file allows you to set common environment variables usable by all scripts in the $JBOSS_HOME/bin directory. You can add the file to $JBOSS_HOME/bin or add the path to the file in the COMMON_CONF environment variable. This functionality does support batch scripts and powershell scripts, with common.conf.bat and common.conf.ps1 respectively.

2.3. MANAGEMENT CLI

Enhancement to the command CLI command
The CLI command command has a new --node-child argument that you can use to edit the properties or manage the operations of a specific child node.

NOTE
Before you use the --node-child argument, check that the child node exists in the management model.

Use the command add --node-child --help CLI command to view a description of the --node-child argument.

New role decoder added to the elytron subsystem
In JBoss EAP 7.4, you can use the management CLI to add the source-address-role-decoder role decoder to the elytron subsystem. By configuring this role decoder in the mappers element, you can gain additional information from a remote client when making authorization decisions.

You can configure the following attributes for source-address-role-decoder:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pattern</td>
<td>A regular expression that specifies the IP address of a remote client or the IP addresses of remote clients to match.</td>
</tr>
<tr>
<td>source-address</td>
<td>Specifies the IP address of the remote client.</td>
</tr>
<tr>
<td>roles</td>
<td>Provides the list of roles to assign to a user if the IP address of the remote client matches the values specified in the pattern attribute or the source-address attribute.</td>
</tr>
</tbody>
</table>
Exposing runtime statistics for managed executor services

In the previous JBoss EAP release, runtime statistics were not available for managed executor services in the ee subsystem.

You can now monitor the performance of managed executor services by viewing the runtime statistics generated with the new management CLI attributes. The following management CLI attributes have been added:

- **active-thread-count**: the approximate number of threads that are actively executing tasks
- **completed-task-count**: the approximate total number of tasks that have completed execution
- **hung-thread-count**: the number of executor threads that are hung
- **max-thread-count**: the largest number of executor threads
- **current-queue-size**: the current size of the executor’s task queue
- **task-count**: the approximate total number of tasks that have ever been submitted for execution
- **thread-count**: the current number of executor threads

Terminating hung tasks

You can now manually attempt to terminate hung tasks in the EE subsystem. To do this, run the following command:

```
/subsystem=ee/managed-executor-service=default:terminate-hung-tasks()
```

A new attribute, **hung-task-termination-period**, is added to the managed-executor-service

You can now automatically attempt to terminate hung tasks in the EE subsystem. A new attribute, **hung-task-termination-period**, is added to the managed-scheduled-executor-service resources to facilitate this.

- **hung-task-termination-period**: the period, in milliseconds, for attempting hung tasks automatic termination, by cancelling such tasks, and interrupting their executing threads. If value is 0, which is the default, hung tasks are never cancelled.

Using property replacement for permissions files

Users upgrading from JBoss EAP 6 to JBoss EAP 7 were unable to migrate file permissions in the Java policy file to the permissions.xml or jboss-permissions.xml files. It was not possible to use property replacement to migrate file permissions in the permissions.xml and jboss-permissions.xml files.

You can now use property replacement for the permissions.xml and jboss-permissions.xml files.

The property replacement for jboss-permissions.xml and permissions.xml files can be enabled or disabled using the jboss-descriptor-property-replacement and spec-descriptor-property-replacement attributes in the ee subsystem.

Configuring RESTEasy parameters

You can now use the JBoss EAP management CLI to change the settings for RESTEasy parameters. A global change applies the updated settings to new deployments as web.xml context parameters.

You can modify the settings of a parameter by using the :write-attribute operation with the /subsystem=jaxrs resource in the management CLI. For example:
NOTE
When you change the settings of a parameter, the updated settings only apply to new deployments. Restart the server to apply the new settings to current deployments.

See the RESTEasy Configuration Parameters table for details about RESTEasy elements.

Configuring RESTEasy providers
In RESTEasy, certain built-in providers are enabled by default. You can now use the new RESTEasy parameter `resteasy.disable.providers` in the JBoss EAP management CLI to disable specific built-in providers.

The following example demonstrates how to disable the built-in provider `FileProvider`:

```xml
/subsystem=jaxrs:write-attribute(name=resteasy-disable-providers, value=[org.jboss.resteasy.plugins.providers.FileProvider])
```

You can use the `resteasy.disable.providers` parameter with the pre-existing parameter `resteasy.use.builtin.providers` to customize a specific provider configuration that applies to all new deployments.

NOTE
When you change the settings of the `resteasy.disable.providers` parameter, the updated settings only apply to new deployments. Restart the server to apply the new settings to current deployments.

2.4. MANAGEMENT CONSOLE

New role decoder added to the `elytron` subsystem
In JBoss EAP 7.4, you can use the management console to add the `source-address-role-decoder` role decoder to the `elytron` subsystem. By configuring this role decoder in the `mappers` element, you gain additional information from a remote client when you make authorization decisions.

You can configure the following attributes for `source-address-role-decoder`:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pattern</code></td>
<td>A regular expression that specifies the IP address of a remote client or the IP addresses of remote clients to match.</td>
</tr>
<tr>
<td><code>source-address</code></td>
<td>Specifies the IP address of the remote client.</td>
</tr>
<tr>
<td><code>roles</code></td>
<td>Provides the list of roles to assign to a user if the IP address of the remote client matches the values specified in the <code>pattern</code> attribute or the <code>source-address</code> attribute.</td>
</tr>
</tbody>
</table>
2.5. LOGGING

The Apache Log4j2 API

In JBoss EAP 7.4, you can use an Apache Log4j2 API instead of an Apache Log4j API to send application logging messages to your JBoss LogManager implementation.

The JBoss EAP 7.4 release supports the Log4j2 API, but the release does not support the Apache Log4j2 Core implementation, org.apache.logging.log4j:log4j-core, or its configuration files.

2.6. EJB3 SUBSYSTEM

Default global stateful session bean timeout value in the ejb3 subsystem

In the ejb3 subsystem, you can now configure a default global timeout value for all stateful session beans (SFSBs) that are deployed on your server instance by using the default-stateful-bean-session-timeout attribute. This attribute is located in the JBoss EAP server configuration file. You can configure the attribute using the Management CLI.

Attribute behavior varies according to the server mode. For example:

- When running in the standalone server, the configured value gets applied to all SFSBs deployed on the application server.
- When running in the managed domain, all SFSBs that are deployed on server instances within server groups receive concurrent timeout values.

**NOTE**
When you change the global timeout value for the attribute, the updated settings only apply to new deployments. Reload the server to apply the new settings to current deployments.

By default, the attribute value is set at -1 milliseconds, which means that deployed SFSBs are configured to never time out. However, you can configure two other types of valid values for the attribute, as follows:

- When the value is 0, SFSBs are eligible for immediate removal by the ejb container.
- When the value is greater than 0, the SFSBs remain idle for the specified time before they are eligible for removal by the ejb container.

You can still use the pre-existing @StatefulTimeout annotation or the stateful-timeout element, which is located in the ejb-jar.xml deployment descriptor, to configure the timeout value for an SFSB. However, setting such a configuration overrides the default global timeout value to the SFSB.

Forcing Jakarta Enterprise Beans timer refresh in database-data-store

You can now set the wildfly.ejb.timer.refresh.enabled flag using the EE interceptor. When an application calls the TimerService.getAllTimers() method, JBoss EAP checks this flag. If this flag is set to true, JBoss EAP refreshes the Jakarta Enterprise Beans timers from database before returning the result.

In the previous JBoss EAP releases, the Jakarta Enterprise Beans timer reading could be refreshed in a database using the refresh-interval attribute found in database-data-store. Users could set the refresh-interval attribute value in milliseconds to refresh the Jakarta Enterprise Beans timer reading.
For information about Jakarta Enterprise Beans clustered database backed-timers, see Jakarta Enterprise Beans clustered database timers in the Developing EJB Applications guide.

Access to runtime information from Jakarta Enterprise Beans
With JBoss EAP 7.4, you can access runtime data for Jakarta Enterprise Beans. Stateful session beans, stateless session beans, and singleton beans each return different runtime information. For example, the following command returns runtime data for a stateless session bean:

```
/deployment=ejb-management.jar/subsystem=ejb3/stateless-session-bean=ManagedStatelessBean:read-resource(include-runtime)
```

Dynamic discovery of Jakarta Enterprise Beans over HTTP
With JBoss EAP 7.4, you can use dynamic discovery of Jakarta Enterprise Beans over HTTP. To use this capability, add a configuration similar to the following to the `ejb-remote` profile:

```
<remote>
  <channel-creation-options>
    <option name="MAX_OUTBOUND_MESSAGES" value="1234" type="remoting"/>
  </channel-creation-options>
  <profiles>
    <profile name="my-profile">
      <remote-http-connection name="ejb-http-connection" url="http://127.0.0.1:8180/wildfly-services"/>
    </profile>
  </profiles>
</remote>
```

Global configuration of compression for remote Jakarta Enterprise Beans calls
With JBoss EAP 7.4, you can configure compression of calls to remote Jakarta Enterprise Beans globally. To configure compression globally on a stand-alone client, specify the `default.compression` property in the `jboss-ejb-client.properties` file. To configure compression globally on a server, include the `default-compression` attribute in the `<client-context>` element in the jboss-ejb-client.xml descriptor file in the application deployment unit.

```
<jboss-ejb-client xmlns="urn:jboss:ejb-client:1.4">
  <client-context default-compression="5">
    <profile name="example-profile"/>
  </client-context>
</jboss-ejb-client>
```

New attribute for setting the principal propagation behavior in Elytron
JBoss EAP 7.4 adds the `legacy-compliant-principal-propagation` attribute to the `application-security-domain` element of the `ejb3` subsystem. You can use the optional `legacy-compliant-principal-propagation` attribute to set the principal propagation behavior of your EJB application that uses Elytron security.

JBoss EAP 7.4 sets the default value of the attribute as `true`, which configures the principal propagation behavior as legacy security subsystem compliant.

If you configure the attribute to `false`, Elytron provides any local unsecured EJBs that have no incoming run as identity with an anonymous principal. This configuration complies with Elytron's previous behavior.

2.7. HIBERNATE
Configuring the `wildfly.jpa.skipquerydetach` persistence unit property
You can configure the `wildfly.jpa.skipquerydetach` persistence unit property from the `persistence.xml` file of a container-managed persistence context.

The default value for `wildfly.jpa.skipquerydetach` is `false`. Use this setting to set a transaction-scoped persistence context to immediately detach query results from an open persistence context.

Configure `wildfly.jpa.skipquerydetach` as `true`, to set a transaction-scoped persistence context to detach query results when a persistence context is closed. This enables a non-standard specification extension.

For applications that have the non-standalone specification extension `jboss.as.jpa.deferdetach` set as `true`, you can also set `wildfly.jpa.skipquerydetach` as `true`.

2.8. WEB SERVICES

Integrating Elytron with web services clients
You can now configure web services clients to use the Elytron client configuration to obtain its credentials, authentication method, and SSL context.

When you use JBossWS API to assign any configuration properties to the web services client, then the username, password, and SSL context from the Elytron client are also loaded and configured. The following authentication methods are configurable:

- UsernameToken Profile authentication
- HTTP Basic authentication
- TLS protocol

You can use the `<webservices/>` element in `wildfly-config.xml` to specify that the credentials are for either HTTP Basic authentication, UsernameToken Profile authentication or both.

Ability for RESTEasy 3.x to access all standard MicroProfile ConfigSources
RESTEasy 3.x can now access all standard MicroProfile ConfigSources. The following additional ConfigSources are also added to RESTEasy 3.x:

- `servlet init-params` (ordinal 60)
- `filter init-params` (ordinal 50)
- `servlet context-params` (ordinal 40)

Previously, these capabilities were only included in RESTEasy 4.x. With this update, RESTEasy can access configuration parameters with or without the MicroProfile ConfigSources. In the absence of a MicroProfile Config implementation, RESTEasy falls back to the older method of gathering parameters from ServletContext parameters and init parameters.

Configuring SameSite cookie attribute
You can now configure the SameSite attribute for cookies in the current JBoss EAP release with a `samesite-cookie` predicated handler in the `undertow` subsystem. With this handler, you can update your server configuration without having to change your applications. This enhancement supports changes to the processing of cookies that were recently implemented in major web browsers to improve security.

Configuring Eclipse MicroProfile REST client API in resteasy CDI modules
The Eclipse MicroProfile REST client API is now an optional dependency that you can configure in resteasy CDI modules.

2.9. MESSAGING

Duplicate messages on the JMS core bridge
On rare instances for a server with an overloaded target queue, sending large messages over the JMS core bridge might cause duplication of your messages.

Ability to pause a topic
With JBoss EAP 7.4, you can pause a topic in addition to pausing a queue. When you pause a topic, JBoss EAP receives messages but does not deliver them. When you resume the topic, JBoss EAP delivers the messages. To pause a topic, issue a command similar to the following example:

```
/subsystem=messaging-activemq/server=default/jms-topic=topic:pause()
```

To resume a topic, issue a command similar to the following example:

```
/subsystem=messaging-activemq/server=default/jms-topic=topic:resume()
```

Ability to detect network isolation of broker
You can now ping a configurable list of hosts to detect network isolation of the broker. You can use the following parameters to configure this functionality:

- `network-check-NIC`
- `network-check-period`
- `network-check-timeout`
- `network-check-list`
- `network-check-URL-list`
- `network-check-ping-command`
- `network-check-ping6-command`

For example, to check the network status by pinging the IP address `10.0.0.1`, issue the following command:

```
/subsystem=messaging-activemq/server=default:write-attribute(name=network-check-list, value="10.0.0.1")
```

`call-timeout` attribute
The `call-timeout` attribute on the JMS core bridge is configurable as a part of ActiveMQ Artemis. In this release, you are able to configure the `call-timeout` variable in EAP itself with the management API.

Red Hat AMQ connection pools
Red Hat AMQ recently began supporting connection pools in addition to single thread DB connections. With JBoss EAP 7.4, you can now use a connection pool when using Red Hat AMQ with JBoss EAP.

2.10. SCRIPTS
New environment variable for starting your server
You can now add the `MODULE_OPTS` environment variable to the script configuration files in your JBoss EAP 7.4 instance.

In a standalone server, use the following files:

- On RHEL, the startup script uses `EAP_HOME/bin/standalone.conf` file.
- On your Windows server, at the command prompt, use the `EAP_HOME\bin\standalone.bat` file.
- On your Windows server, at the PowerShell, use the `EAP_HOME\bin\standalone.ps1` file.

For servers in a domain, you can add the `module-options` attributes to a host JVM configuration or a server’s JVM configuration.

The `MODULE_OPTS` environment variable affects the entire server. For example, if you have a Java agent that requires logging, set the value of `MODULE_OPTS` to `-javaagent:my-agent.jar`. This will initialize your agent after you configure logging.

2.11. OPENSHIFT

Providing custom Galleon feature-pack support to your JBoss EAP S2I image
You can use three new environment variables to provide custom Galleon feature-pack support for your JBoss EAP S2I image. You can use the environment variables outlined in the following table during your S2I build phase:

Table 2.1. Custom Galleon feature-pack environment variables

<table>
<thead>
<tr>
<th>Environment variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GALLEON_DIR=&lt;PATH&gt;</td>
<td><code>&lt;PATH&gt;</code> is the relative directory to the application root directory that contains your optional Galleon custom content. Directory defaults to <code>galleon</code>.</td>
</tr>
<tr>
<td>GALLEON_CUSTOM_FEATURE_PACKS_MAVEN_REPO=&lt;PATH&gt;</td>
<td><code>&lt;PATH&gt;</code> is the absolute path to a Maven local repository directory that contains custom feature-packs. Directory defaults to <code>galleon/repository</code>.</td>
</tr>
<tr>
<td>GALLEON_PROVISION_FEATURE_PACKS= &lt;LIST_OF_GALLEON_FEATURE_PACKS&gt;</td>
<td><code>&lt;LIST_OF_GALLEON_FEATURE_PACKS&gt;</code> is a comma-separated list of your custom Galleon feature-packs identified by Maven coordinates. The listed feature-packs must be compatible with the version of the JBoss EAP 7.4 server present in the builder image. You can use the <code>GALLEON_PROVISION_LAYERS</code> environment variable to set the Galleon layers, which were defined by your custom feature-packs, for your server.</td>
</tr>
</tbody>
</table>

Read-only server configuration directory
JBoss EAP supports a read-only server configuration directory. You can use the `--read-only-server-config` command line parameter to lock down the server configuration when the server configuration directory is a read-only directory. This functionality is available only when running JBoss EAP as a standalone server.

Instructions to deploy JBoss EAP quickstarts on OpenShift
For a JBoss EAP release, all OpenShift-compatible quickstarts now include instructions to deploy JBoss EAP quickstarts on OpenShift. The readme.html file of the quickstarts include the following sections:

- Getting Started with OpenShift
- Prepare OpenShift for Quickstart Deployment
- Import the Latest JBoss EAP for OpenShift Image Streams and Templates
- Deploy the JBoss EAP for OpenShift Source-to-Image (S2I) Quickstart to OpenShift
- OpenShift Post Deployment Tasks

New Galleon layer for the Distributable Web subsystem
JBoss EAP provides the web-passivation layer to supply the distributable-web subsystem configured with a local web container cache. The web-passivation layer is a decorator layer.

2.12. RED HAT CODEREADY WORKSPACES (CRW)

Red Hat CodeReady Workspaces supports JBoss EAP 7.4 development files
You can use a JBoss EAP 7.4 development file, YAML file, to define a JBoss EAP development environment on CRW. You can download example JBoss EAP 7.4 development files from the jboss-eap-quickstarts GitHub web page.

A development file includes the following components:

- A browser IDE configuration
- A list of predefined commands
- The application runtime environment
- The location of the repository that you must clone

On CRW, you can choose one of the following ways to create a JBoss EAP 7.4 workspace environment:

- Copy and paste the URL of a JBoss EAP development file directly into the Devfile section of the Get Started page on your CRW dashboard. You must select the Load devfile button to add the development file to your CRW dashboard.

- Open your CRW instance on OpenShift and enter the URL of a JBoss EAP development file in the Devfile tab on the Workspace menu. Save the development file and then restart your CRW instance.

IMPORTANT
If you want to use a Java 8 development file in your JBoss EAP 7.4 workspace environment, do not install a Java 11 plug-in because it conflicts with the Java 8 plug-in.

Additional resources
- For information on how to download example JBoss EAP 7.4 development files, go to the kitchensink-jsp subdirectory in the jboss-eap-quickstarts directory on the GitHub web page.
For more information about downloading and installing the latest version of CRW that is compatible with JBoss EAP 7.4, see Installing CodeReady Workspaces in the CRW Installation Guide.

For more information about configuring CRW, see Configuring a CodeReady Workspaces 2.9 workspace in the CRW End-user Guide.
CHAPTER 3. UNSUPPORTED FUNCTIONALITY

3.1. UNSUPPORTED FEATURES

Support for some technologies are removed due to the high maintenance cost, low community interest, and better alternative solutions.

Platforms and features
JBoss EAP deprecated the following platforms in version 7.1. These platforms are not tested in JBoss EAP 7.4.

- Oracle Solaris 10 on x86_64
- Oracle Solaris 10 on SPARC64
- Oracle Solaris 11 on x86_64
- Oracle Solaris 11 on SPARC64

JBoss EAP 7.4 does not include the Wildfly SSL natives for these platforms. As a result, SSL operations in Oracle Solaris platforms might be slower than they were on previous versions of JBoss EAP.

Keystore defect with Java jdk8u292-b10
If you’re running JBoss EAP on Java jdk8u292-b10 and using a legacy security realm or an Elytron Lightweight Directory Access Protocol (LDAP) keystore, you cannot use a Public-Key Cryptography Standards (PKCS) #12 keystore. The workaround is to configure your instance of JBoss EAP to use a stronger default key protection algorithm for PKCS #12 keystores. Other Elytron keystore types are not affected by this defect.

RESTEasy parameters
RESTEasy provides a Servlet 3.0 ServletContainerInitializer integration interface that performs an automatic scan of resources and providers for a servlet. Containers can use this integration interface to start an application. Therefore, use of the following RESTEasy parameters is no longer supported:

- resteasy.scan
- resteasy.scan.providers
- resteasy.scan.resources

MicroProfile capabilities
The following MicroProfile capabilities that were included as technical preview in JBoss EAP 7.3 are not included in JBoss EAP 7.4 or in future versions:

- MicroProfile Config
- MicroProfile REST client
- MicroProfile Health
  JBoss EAP no longer includes the microprofile-smallrye-health subsystem, so application healthiness checks are no longer available. JBoss EAP continues to include healthiness check for the server runtime.
- MicroProfile Metrics
JBoss EAP no longer includes the `microprofile-smallrye-metrics` subsystem, so application metrics are no longer available. JBoss EAP continues to include endpoints for JVM and server metrics.

- MicroProfile OpenTracing
  MicroProfile OpenTracing is now part of the `observability` decorator layer.

These capabilities are now part of the JBoss EAP Expansion Pack (JBoss EAP XP). Install JBoss EAP XP for full MicroProfile support in JBoss EAP.

For complete information about support for MicroProfile and JBoss EAP XP, see the JBoss EAP XP lifecycle and support policies page.

**Red Hat JBoss Operations Network**
Using Red Hat JBoss Operations Network (JON) for JBoss EAP management is deprecated since JBoss EAP version 7.2. For JBoss EAP 7.4, support for Red Hat JON for JBoss EAP management is deprecated.

**MS SQL Server 2017**
MS SQL Server 2017 is not supported in JBoss EAP 7.4.

### 3.2. DEPRECATED FEATURES

Some features have been deprecated with this release. This means that no enhancements will be made to these features, and they may be removed in the future, usually the next major release.

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.

For details of which features have been deprecated, see the JBoss Enterprise Application Platform Component Details located on the Red Hat Customer Portal.

**Platforms and features**
Support for the following platforms and features is deprecated:

**Eclipse MicroProfile REST Client API**
The Eclipse MicroProfile REST Client API is deprecated from the `jaxrs` subsystem.

**OpenShift Container Platform 3.11**
OpenShift Container Platform (OCP) 3.11 is deprecated for JBoss EAP 7.4.

**Operating systems**
- Microsoft Windows Server on i686
- Red Hat Enterprise Linux (RHEL) 6 on i686

**NOTE**
Although support for these platforms was deprecated in a previous JBoss EAP release, some artifacts and resources linked to these platforms were not removed, such as the `wildfly-openssl` native library binding. For Red Hat JBoss Enterprise Application Platform 7.4, those artifacts and resources have been removed.
Databases and database connectors

- IBM DB2 11.1
- PostgreSQL / EnterpriseDB 11
- MariaDB 10.1
- MS SQL 2017

Lightweight Directory Access Protocol (LDAP) servers

- Red Hat Directory Server 10.0
- Red Hat Directory Server 10.1

Spring BOM
The following Spring BOM that is located in the Red Hat Maven repository is now deprecated:

- jboss-eap-jakartaee8-with-spring4

Although Red Hat tests that Spring applications run on Red Hat JBoss Enterprise Application Platform 7.4, you must use the latest version of the Spring Framework and its BOMs (for example, x.y.z.RELEASE) for developing your applications on JBoss EAP 7.4.

For more information about versions of the Spring Framework, see Spring Framework Versions on GitHub.

BOMs
The existing BOMs are deprecated with a view to providing BOMs (perhaps including some of the existing ones) relevant to the functionality in the next major version of JBoss EAP.

Java Development Kits (JDKs)

- JDK 8
- JDK 11

NOTE
In future JBoss EAP releases, Java SE requirements will be reevaluated based on the industry (for example, Jakarta EE 10+, MicroProfile and so on) and market needs.

JBoss EAP OpenShift templates
JBoss EAP templates for OpenShift are deprecated.

eap74-beta-starter-s2i.json and eap73-third-party-db-s2i.json templates
The eap74-beta-starter-s2i.json and eap74-beta-third-party-db-s2i.json templates are deprecated and are removed in JBoss EAP 7.4.0.GA.

Legacy security subsystem
The org.jboss.as.security extension and the legacy security subsystem it supports are now deprecated. Migrate your security implementations from the security subsystem to the elytron subsystem.

PicketLink
The org.wildfly.extension.picketlink extension, and the picketlink-federation and picketlink-identity-management subsystems this extension supports, are now deprecated. Migrate your single sign-on implementation to Red Hat Single Sign-On.
**PicketBox**
The PicketBox-based security vault, including access by using the legacy `security` subsystem and the `core-service=vault` kernel management resources, is now deprecated in this release.

**Managed domain support for previous versions of JBoss EAP**
Support for hosts running JBoss EAP 7.3 and earlier versions in a JBoss EAP 7.4 managed domain is deprecated. Migrate the hosts in your managed domains to JBoss EAP 7.4.

**Server configuration files using namespaces from JBoss EAP 7.3 and earlier**
Using server configuration files (`standalone.xml`, `host.xml`, and `domain.xml`) that include namespaces from JBoss EAP 7.3 and earlier is deprecated in this release. Update your server configuration files to use JBoss EAP 7.4 namespaces.

**JBoss EAP Server Side JavaScript support**
Previously, JBoss EAP Server Side JavaScript support was offered as a Technology Preview. It is now deprecated in this release.

**Agroal subsystem**
The `datasources-agroal` subsystem is deprecated.

**Codehaus Jackson**
The Codehaus Jackson 1.x module, which is currently unsupported, is deprecated in JBoss EAP 7.4.

**application-security-domain resources**
The `application-security-domain` resources in `ejb3` and `undertow` subsystems are deprecated.

**Clustering subsystems**
The following resources in the clustering subsystems are deprecated:

- The `infinispan` subsystem
  ```
  /subsystem=infinispan/remote-cache-container=*/component=transaction
  /subsystem=infinispan/remote-cache-container=*/near-cache=*  
  ```
- The `jgroups` subsystem
  ```
  /subsystem=jgroups/stack=*/protocol=S3_PING
  /subsystem=jgroups/stack=*/protocol=GOOGLE_PING  
  ```
- The `modcluster` subsystem

**Salted Challenge Response Authentication Mechanism**
The following Salted Challenge Response Authentication Mechanisms (SCRAMs) and their channel-binding variants are deprecated:

- SCRAM-SHA-512
- SCRAM-SHA-384

**Quickstarts**
The existing Quickstarts are deprecated with a view to providing Quickstarts (perhaps including some of the existing ones), relevant to the functionality in the next major version of JBoss EAP.
**Hibernate ORM 5.1**
The Hibernate ORM 5.1 native API bytecode transformer has always been deprecated since it was originally introduced.

**HornetQ messaging client**
The HornetQ messaging client is deprecated.
CHAPTER 4. RESOLVED ISSUES

See Resolved Issues for JBoss EAP 7.4 to view the list of critical issues we have resolved for this release.

Additionally, be aware of the following:

- After completing source-to-image builds, OpenShift now clears the source directory (/tmp/src). As a result of this change, built images should be smaller.
CHAPTER 5. FIXED CVES

JBoss EAP 7.4 includes fixes for the following security-related issues:

- **CVE-2020-14317**: WildFly: With JBoss EAP, the race condition on process identification number (PID) files lets local users terminate arbitrary processes.
CHAPTER 6. KNOWN ISSUES

See Known Issues for JBoss EAP 7.4 to view the list of known issues for this release.

6.1. CHANGED BEHAVIORS FOR JBOSS EAP 7.4

Setting OPENSHIFT_DNS_PING_SERVICE_NAME to an empty value results in boot error
Do not set OPENSHIFT_DNS_PING_SERVICE_NAME to an empty value. A boot error occurs and clustering is disabled.

Unpredictable web session expiration
Previously, JBoss EAP mistakenly recalculated some web session timeouts, and these sessions did not expire as specified in the web application configuration file (such as web.xml, jboss-web.xml or jboss-all.xml). JBoss EAP no longer performs this mistaken calculation, so web sessions now expire as specified in the application configuration.

Memory leaks in distributed JSF applications when caching managed beans in a WebInjectionContainer
JBoss EAP cluster membership changes, such as starting or stopping a server, can cause the events that correspond to a given session to resume on a different cluster member than the one that last handled those events. Specifically, org.jboss.as.web.common.WebInjectionContainer caches references to all managed beans and their references so that it can call ManagedReference.release, which causes a memory leak. This issue affects distributed Jakarta Server Faces (JSF) applications that use the JBoss EAP high-availability (HA) server configuration. References to session-scoped beans can persist after the associated HTTP session expires, if a different cluster member handles that expiration. As a workaround, change the distributable-web subsystem like in the following example:

/subsystem=distributable-web/infinispan-session-management=default/affinity=local:add

Java.lang.NullPointerException error when using ibm-java-1.8 and Bouncy Castle
If you’re directly or indirectly using the Bouncy Castle provider with IBM JDK 1.8 on JBoss EAP, you might get the following error:

Caused by: java.lang.NullPointerException
at org.bouncycastle.jcajce.provider.asymmetric.rsa.BCRSAPrivateKey.getAlgorithm(BCRSAPrivateKey.java:79)
at com.ibm.crypto.provider.bf.supportsParameter(Unknown Source)
at javax.crypto.Cipher.a(Unknown Source)
at javax.crypto.Cipher.init(Unknown Source)
at javax.crypto.Cipher.init(Unknown Source)
at org.bouncycastle.operator.jcajce.JceAsymmetricKeyUnwrapper.generateUnwrappedKey(JceAsymmetricKeyUnwrapper.java:109)
at org.bouncycastle.cms.jcajce.JceKeyTransRecipient.extractSecretKey(JceKeyTransRecipient.java:208)
at org.bouncycastle.cms.jcajce.JceKeyTransEnvelopedRecipient.getRecipientOperator(JceKeyTransEnvelopedRecipient.java:26)
at org.bouncycastle.cms.KeyTransRecipientInformation.getRecipientOperator(RecipientInformation.java:48)
at org.bouncycastle.cms.RecipientInformation.getContentStream(RecipientInformation.java:169)
To work around this issue, modify your JBoss EAP `module.xml` structure similarly to that of the WFLY-14688 diff, which you can access in the Additional resources section.

Additional resources

- For more information about working around this issue, see WFLY-14688 diff.
- For more information about Bouncy Castle cryptography APIs, see bouncycastle.org.