



Red Hat JBoss Enterprise Application Platform 7.2

7.2.0 リリースノート

Red Hat JBoss Enterprise Application Platform 7.2 向け

Red Hat JBoss Enterprise Application Platform 7.2 7.2.0 リリースノート

Red Hat JBoss Enterprise Application Platform 7.2 向け

法律上の通知

Copyright © 2019 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本リリースノートには、Red Hat JBoss Enterprise Application Platform 7.2 に関する重要な情報が含まれています。

目次

第1章 サポートされる構成	5
第2章 新機能および改良された機能	6
2.1. JAVA EE 8	6
Java EE 8 のサポート	6
Java EE 8 のセキュリティーサポート	6
2.2. セキュリティー	6
Red Hat シングルサインオンによる JBoss EAP 管理コンソールのセキュア化	6
SSL の必要時にサーバーが非 SSL IIOOP ソケットをブロック	6
BouncyCastle プロバイダーを使用した FIPS 140-2 対応の暗号化	6
BouncyCastle プロバイダーを使用した FIPS 140-2 対応クレデンシャルストアの定義	6
CLI セキュリティーコマンドを使用した管理インターフェースの SASL 認証の有効化	7
CLI セキュリティーコマンドを使用した HTTP 認証の有効化	7
2.3. サーバー管理	7
Git を使用した設定データの管理	7
サーバーグループの Kill Servers 操作	7
2.4. 管理 CLI	7
キーボード操作ショートカット	7
HTTP 管理 API の出力の生成	8
管理 CLI 出力のスクロール	8
管理 CLI 出力の検索	8
CLI のカラー出力	8
進化したヘルプ	8
必須属性の表示	8
複数ページの出力の表示	8
for-done 制御フローの使用	8
JSON 形式の操作応答の出力	9
出力のリダイレクト	9
統一された deployment コマンド	9
2.5. 管理コンソール	9
トポロジービュー	9
Breadcrumb バー	9
ナビゲーションの改良	9
デプロイメントの改良	9
管理コンソールの SSL ウィザードの有効化	9
Undertow HTTPS リスナーの SSL ウィザードの有効化	10
ロギングプロファイルの設定	10
Security Manager サブシステムの設定	10
Elytron コンポーネントの設定	10
セッションの詳細表示	10
アクティブなセッションの無効化	11
散在 (scattered) キャッシュの設定	11
その他のサブシステム設定	11
追加のサブシステム監視サポート	11
2.6. WEB SERVER	12
Undertow バイトバッファプール	12
デフォルトのクッキーバージョンの設定	12
URL でエスケープ処理されていない文字を使用可能	12
PROXY プロトコル	12
Forwarded HTTP 拡張	12
セッションマネージャーの操作	12

2.7. IO	13
新しい worker 属性	13
2.8. ロギング	13
ソケットログハンドラー	13
JSON および XML フォーマッター	13
2.9. トランザクション	13
新しい maximum-timeout トランザクションマネージャー属性	13
2.10. DATASOURCES	14
JDBC ドライバーのデータソースクラスプロパティの取得	14
2.11. EJB	14
HTTP/HTTPS 上の EJB および JNDI と HTTP ロードバランサー	14
コンテキストデータの EJB が クライアントへの返信	14
2.12. JSF	14
JSF デプロイメントでの DOCTYPE 宣言の拒否	14
2.13. HIBERNATE	14
Hibernate ORM 5.1 から Hibernate ORM 5.3 へのアップグレード	14
Hibernate Validator 5.3.x から Hibernate Validator 6.0.x へのアップグレード	14
2.14. クラスタリング	15
改良された execute メソッドによって使用される CompletableFuture	15
2.15. INFINISPAN	15
HotRod クライアントインジェクション	15
非ブロッキングの初期状態遷移	15
散在 (scattered) キャッシュモード	15
リモートキャッシュストアを使用した HTTP セッションの外部的化	15
2.16. WEB サービス	16
JSON バインディングの Java API	16
非同期 HTTP リクエストの処理	16
カスタム RESTEasy アノテーション	16
ParamConverter 機能の拡張	16
リソースメソッドアルゴリズムの切り替え	16
RESTEasy サービスプロバイダーインターフェース	17
HTTP リダイレクトの JAX-RS クライアントサポート	17
2.17. メッセージング	17
IBM MQ リソースアダプター	17
JDBC データベースを使用したメッセージングジャーナルの永続化	17
メッセージング JDBC 永続ストアの HA トポロジーのサポート	17
リモート Red Hat AMQ 7 メッセージングブローカーへの接続の簡略化	17
統合された Artemis リソースアダプターを使用した Red Hat AMQ への接続	17
2.18. OPENSIFT	18
JBoss EAP でネイティブに統合された KUBE_PING	18
2.19. モジュール	18
事前定義されたモジュール	18
2.20. クイックスタートおよび BOM	18
アプリケーションの開発に使用可能な JBoss EAP BOM	18
第3章 テクノロジーレビュー	19
新しい Agroal データソースサブシステム	19
MicroProfile REST クライアント	19
非同期リクエスト処理および反応的戻り値の型の RESTEasy サポートの拡張	19
Eclipse MicroProfile Config	19
Eclipse MicroProfile OpenTracing	20
Eclipse MicroProfile Health	20

第4章 サポートされない機能および非推奨の機能	21
4.1. サポートされない機能	21
プラットフォームおよび機能	21
メッセージング (ActiveMQ Artemis)	22
RPM current リポジトリ	22
クイックスタート	22
4.2. 非推奨の機能	23
IO サブシステム	23
キャッシュストア	23
Red Hat JBoss Operations Network	23
プラットフォームおよび機能	23
ドキュメント	24
第5章 解決した問題	25
第6章 修正された CVE	26
第7章 既知の問題	27

第1章 サポートされる構成

以下の構成が JBoss EAP 7.2 で新たにサポートされます。

Java 仮想マシン

- OpenJDK 8 for Windows Server
- OpenJDK 11 for Windows Server and Red Hat Enterprise Linux ¹
- Oracle JDK 11 ¹

¹ JDK 11 のサポートには、JPMS (Java Platform Module System) ユーザーモジュールのサーバーへのデプロイメントは含まれません。

データベースおよびデータベースコネクター

- IBM DB2 e11.1
- PostgreSQL 10.1
- EnterpriseDB Postgres Plus Advanced Server 10.1

JMS プロバイダーおよびアダプター

- IBM MQ 9

Red Hat Developer Studio

- JBoss EAP 7.2 での Red Hat Developer Studio 12 の使用は認定されています。

JBoss EAP 7.2 で完全サポートされる構成の詳細は、[JBoss Enterprise Application Platform \(EAP\) 7 でサポートされる構成](#) のページを参照してください。

第2章 新機能および改良された機能

2.1. JAVA EE 8

Java EE 8 のサポート

JBoss EAP 7.2 には、以下の Java EE 8 標準のサポートが含まれています。

- [JSR 250](#): Common Annotations 1.3
- [JSR 338](#): JPA 2.2
- [JSR 365](#): CDI 2.0
- [JSR 367](#): JSON-B 1.0
- [JSR 369](#): Servlet 4.0
- [JSR 370](#): JAX-RS 2.1
- [JSR 372](#): JSF 2.3
- [JSR 374](#): JSON-P 1.1
- [JSR 380](#): Bean Validation 2.0
- [JSR 919](#): JavaMail 1.6

Java EE 8 のセキュリティーサポート

Java EE 8 には、認証およびアイデンティティストアの移植可能なプラグインインターフェースである [JSR 375](#) のサポートと、プログラムによるセキュリティーのアクセスポイントを提供する新しい **injectable-type SecurityContext** インターフェースのサポートが含まれます。これらの API に組み込まれた実装を使用するか、カスタム実装を定義することができます。

JBoss EAP 7.2 は [JSR 375](#) をサポートするようになりました。

2.2. セキュリティー

Red Hat シングルサインオンによる JBoss EAP 管理コンソールのセキュア化

Red Hat シングルサインオンを設定して、JBoss EAP 管理コンソールの認証を管理できるようになりました。

詳細は、『[How to Configure Server Security](#)』の「[Secure the Management Console with Red Hat Single Sign-On](#)」を参照してください。

SSL の必要時にサーバーが非 SSL IIOP ソケットをブロック

IIOP サブシステムで `server-requires-ssl` 属性が `true` に設定されている場合、サーバーが SSL でないソケットへの接続をブロックするようになりました。

BouncyCastle プロバイダーを使用した FIPS 140-2 対応の暗号化

`elytron` サブシステムを使用すると、BouncyCastle プロバイダーを使用して JBoss EAP の FIPS 対応インスタンスを設定できます。完全な手順は『[How to Configure Server Security](#)』の「[Enable FIPS 140-2 Cryptography for SSL/TLS Using BouncyCastle](#)」を参照してください。

BouncyCastle プロバイダーを使用した FIPS 140-2 対応クレデンシャルストアの定義

BouncyCastle プロバイダーを使用して、FIPS 対応のクレデンシャルストアを取得できます。これらのクレデンシャルストアは、以下の方法のいずれかを使用して定義できます。

- **elytron** サブシステムから直接クレデンシャルストアを定義する手順は、「[Define a FIPS 140-2 Compliant Credential Store Using the BouncyCastle Providers](#)」を参照してください。
- WildFly Elytron ツールを使用してクレデンシャルストアをオフラインで定義する手順は、「[Create and Modify Credential Stores Offline with the WildFly Elytron Tool](#)」を参照してください。

CLI セキュリティーコマンドを使用した管理インターフェースの SASL 認証の有効化
security enable-sasl-management CLI コマンドを使用して、SASL 認証を管理インターフェースに対して有効にできるようになりました。このコマンドは、認証の設定に必要で存在しないリソースをすべて作成します。

詳細は、『[How to Configure Server Security](#)』の「[Enable SASL Authentication for the Management Interfaces Using the CLI Security Command](#)」を参照してください。

CLI セキュリティーコマンドを使用した HTTP 認証の有効化
security CLI コマンドを使用して、Undertow セキュリティードメインおよび管理インターフェースに対して HTTP 認証を有効化できるようになりました。

- Undertow セキュリティードメインの場合は **security enable-http-auth-http-server** CLI コマンドを使用します。
- 管理インターフェースの場合は **security enable-http-auth-management** CLI コマンドを使用します。

詳細は、『[How to Configure Server Security](#)』の「[Enable HTTP Authentication for Applications Using the CLI Security Command](#)」と「[Enable HTTP Authentication for the Management Interfaces Using the CLI Security Command](#)」を参照してください。

2.3. サーバー管理

Git を使用した設定データの管理

Git を使用してサーバー設定データ、プロパティファイル、およびデプロイメントを管理および永続化できるようになりました。これにより、バージョン履歴を管理できるだけでなく、1 つ以上の Git リポジトリを使用して複数のサーバーおよびノード全体でサーバーやアプリケーションの設定を共有することができます。この機能は、デフォルトの設定ディレクトリーレイアウトを使用するスタンドアロンサーバーでのみ動作します。

詳細は『[Configuration Guide](#)』の「[Using Git to Manage Configuration Data](#)」を参照してください。

サーバーグループの **Kill Servers** 操作

管理対象ドメインのサーバーグループに **kill-servers** 操作を使用できるようになりました。これは、問題によってサーバーグループのすべてのサーバーがハングする場合に便利な操作で、各サーバーに **kill** 操作を実行しなくても、一度にすべてのサーバープロセスを kill することができます。

2.4. 管理 CLI

キーボード操作ショートカット

管理 CLI コマンドの編集時にキーボードのショートカットを使用できるようになりました。使用するキーボードのショートカットは、ご使用のプラットフォームに応じて異なります。サポートされるショートカットのリストは、『[管理 CLI ガイド](#)』の「[キーボード操作ショートカットの使用](#)」を参照してください。

HTTP 管理 API の出力の生成

echo-dmr コマンドは、1つの行に内容を表示する新しい **--compact** 引数を提供します。管理 CLI 起動の引数である **--output-json** と使用すると、HTTP 管理 API によって直接消費される出力を生成することができます。

管理 CLI 出力のスクロール

管理 CLI の出力がターミナルウィンドウよりも長い場合に、コンソール内部を直接スクロールできるようになりました。スクロールホイール、矢印キー、**PgUp**、**PgDn**、**Home**、および **End** キーを使用して出力内を移動できます。

Windows では、この機能は Windows Server 2016 より使用可能です。他のオペレーティングシステムでは問題はありません。

管理 CLI 出力の検索

管理 CLI で複数ページの出力を検索できるようになりました。詳細は『[管理 CLI ガイド](#)』の「[複数ページの出力の検索](#)」を参照してください。

CLI のカラー出力

管理 CLI を設定して、ログメッセージの出力タイプに応じて CLI ログをカラーで出力できるようになりました。使用可能な色やカラー出力を有効または無効にする方法に関する詳細は、『[管理 CLI ガイド](#)』の「[管理 CLI の設定](#)」を参照してください。

進化したヘルプ

管理 CLI の **help** 機能がアップデートされ、ヘルプ情報へのアクセスが容易になりました。**help** コマンドにタブ補完を使用できるようになり、管理 CLI 操作とコマンド動作のヘルプ情報も表示できるようになりました。

[管理 CLI help コマンドの使用](#)に関する詳細は、『[管理 CLI ガイド](#)』を参照してください。

必須属性の表示

管理 CLI でタブ補完を使用すると、現在の操作に必要な属性に ***** が付きます。

```
/subsystem=naming/binding=test:add( [TAB]
!           class           module
binding-type* environment    type
cache      lookup          value
```

上記の例では、**/subsystem=naming/binding=test:add(** の入力後に Tab を押すと使用可能な属性が表示され、この操作に必要な属性は **binding-type** であることが示されます。

複数ページの出力の表示

管理 CLI を対話モードで実行し、操作によって複数のページが出力される場合、コマンドプロセッサは最初のページの最後で画面を停止します。これにより、出力を 1 行または 1 ページごとに確認することができます。複数のページが出力されると、出力の最後に **--More(NNN%)--** という行のテキストが表示されます。

管理 CLI コマンドの実行時に [複数のページが出力](#)された場合は、『[管理 CLI ガイド](#)』で使用できるオプションを確認してください。

for-done 制御フローの使用

管理 CLI で **for-done** 制御フローを使用すると、操作から返されたコレクションでイテレートを行い、コレクションの各項目にコマンドを実行することができます。

詳細は、『[管理 CLI ガイド](#)』の「[for-done 制御フローの使用](#)」を参照してください。

JSON 形式の操作応答の出力

管理 CLI を設定して、操作応答を純粋な JSON 形式で出力することができます。これには、**EAP_HOME/bin/jboss-cli.xml** ファイルで **output-json** 要素を **true** に設定するか、管理 CLI の起動時に **--output-json** を渡します。デフォルトでは、操作応答は DMR 形式で表示されます。

出力のリダイレクト

管理 CLI 操作からターミナルに出力する代わりに、以下の演算子を使用して出力をリダイレクトすることができます。

- **>**: 出力をファイルシステムのファイルに書き込みます。
- **>>**: 出力をファイルシステムのファイルに追加します。
- **|**: 出力を **grep** コマンドにリダイレクトして出力の検索を行います。

詳細は、『[管理 CLI ガイド](#)』の「[出力のリダイレクト](#)」参照してください。

統一された **deployment** コマンド

管理 CLI の **deployment** コマンドを使用すると、統一されたインターフェースを使用してデプロイメントを管理でき、デプロイメントのデプロイ、アンデプロイ、有効化、無効化、またはデプロイメントに関する情報の表示を行うことができます。

詳細は、『[Configuration Guide](#)』の「[Deploy an Application to a Standalone Server Using the Management CLI](#)」および「[Deploy an Application in a Managed Domain Using the Management CLI](#)」を参照してください。

2.5. 管理コンソール

トポロジビュー

管理対象ドメインでは、ドメインのホスト、サーバーグループ、およびサーバーの概要と、各サーバーの状態を確認できるようになりました。これには、**Runtime** タブから **Topology** を選択します。

Breadcrumb バー

リソースを表示するときに上部にある Breadcrumb バーを使用できます。このバーを使用するとリソースの切り替えが容易になります。Breadcrumb バーから別のウィンドウにリソースを開くこともでき、エキスパートモードに切り替えて管理モデルを閲覧することもできます。

ナビゲーションの改良

本リリースには、JBoss EAP のリソースをナビゲートするための新しいインターフェースが導入されました。矢印キーを使用して、リソースファインダーを移動でき、頻繁に使用する項目をリストの上部に固定することができます。また、迅速な検索のために絞り込みを行ったり、プレビューからリソースの主な属性を確認することもできます。

デプロイメントの改良

本リリースでは、管理コンソールを利用したアプリケーションのデプロイおよび管理のサポートが追加されました。ドラッグアンドドロップによるデプロイメントの追加または置換、デプロイメントコンテンツの閲覧によるテキストおよびイメージのプレビュー、デプロイメントのダウンロード、および展開形式のデプロイメントの作成を実行できます。

管理コンソールの **SSL** ウィザードの有効化

管理コンソールによって使用される、HTTP 管理インターフェースの SSL を有効にするためのウィザードが導入されました。このウィザードを使用すると、任意で相互認証のトラストストアを作成でき、以下のキーストアのシナリオを選択できます。

- 証明書ストアを作成し、自己署名証明書を生成する。
- 証明書ストアはすでにファイルシステムにあるが、キースタアの設定がない。
- 有効な証明書ストアを使用するキースタアの設定が存在する。

スタンドアロンサーバーでウィザードにアクセスするには、**Runtime** タブを選択し、該当するサーバーの **表示** をクリックします。**HTTP 管理インターフェース** を選択し、**SSLの有効化** ボタンをクリックします。

管理対象ドメインでウィザードにアクセスするには、**Runtime** タブを選択し、**ホスト** をクリックします。該当するホストを選択して **表示** → **管理インターフェース** → **HTTP** と選択し、**SSLの有効化** ボタンをクリックします。

Undertow HTTPS リスナーの SSL ウィザードの有効化

Undertow HTTPS リスナーの SSL を有効にするためのウィザードが導入されました。このウィザードを使用すると、任意で相互認証のトラストストアを作成でき、以下のキースタアのシナリオを選択できます。

- 証明書ストアを作成し、自己署名証明書を生成する (**管理対象ドメインでは使用不可**)。
- 証明書ストアはすでにファイルシステムにあるが、キースタアの設定がない。
- 有効な証明書ストアを使用するキースタアの設定が存在する。

スタンドアロンサーバーでウィザードにアクセスするには、**Configuration** タブをクリックし、**Subsystems** → **Web (Undertow)** → **Server** と選択し、該当するサーバーの **表示** をクリックします。**Listener** → **HTTPS Listener** と選択し、該当する HTTPS リスナーを選択して **SSLの有効化** ボタンをクリックします。

管理対象ドメインサーバーでウィザードにアクセスするには、**Configuration** タブをクリックし、**Profile** → **Web (Undertow)** → **Server** と選択し、該当するサーバーの **表示** をクリックします。**Listener** → **HTTPS Listener** と選択し、該当する HTTPS リスナーを選択して **SSLの有効化** ボタンをクリックします。

ロギングプロファイルの設定

管理コンソールを使用して、**logging** サブシステムでロギングプロファイルを設定できるようになりました。

Security Manager サブシステムの設定

管理コンソールからの **security-manager** サブシステムの設定がサポートされるようになりました。

Elytron コンポーネントの設定

管理コンソールを使用した以下の Elytron コンポーネントの設定がサポートされるようになりました。

- マップされたロールマッパー
- 認証局アカウント
- カスタムのセキュリティーイベントリスナー

また、管理コンソールの **Runtime** タブから認証局アカウントのアカウントキーを作成、非アクティブ、更新、および変更できるようになりました。

セッションの詳細表示

管理コンソールを使用してデプロイメントの詳細なセッション情報を表示できるようになりました。

Runtime タブから該当するサーバーを選択し、**Web (Undertow)** → **Deployment** と選択して **表示** をクリックします。**Sessions** ページの表には、すべてのアクティブなセッションの ID、作成時間、および最終アクセス時間が表示されます。また、セッションをクリックするとその属性が表示されます。

アクティブなセッションの無効化

管理コンソールを使用してデプロイメントのアクティブなセッションを無効化できるようになりました。

Runtime タブで該当するサーバーを選択し、**Web(Undertow)** → **Deployment** と選択します。デプロイメントを選択し、**表示** をクリックします。**Sessions** ページで無効にするセッション ID を選択し、**Invalidate session** をクリックします。表示されるポップアップウィンドウの **Yes** をクリックしてセッションを無効にします。

散在 (**scattered**) キャッシュの設定

本リリースでは、**infinispan** サブシステムのキャッシュ設定内で設定できる散在 (**scattered**) キャッシュが導入されました。

さらに、散在 (**scattered**) キャッシュは **hotrod** キャッシュストアの使用をサポートします。

その他のサブシステム設定

以下のサブシステムは、**Configuration** タブで使用できるその他の設定オプションのために追加または改良されました。

- MicroProfile Config SmallRye
- EJB
- Infinispan
- JGroups
- JMX
- Messaging (ActiveMQ)
- Resource Adapters
- Security (Legacy)
- Web (Undertow)

追加のサブシステム監視サポート

本リリースでは、**Runtime** タブで使用できる、以下のサブシステムの新規または改良された監視サポートが提供されます。

- Batch (JBeret)
- Datasources
- JNDI
- EJB
- IO
- JAX-RS

- Messaging (ActiveMQ)
- Transaction
- Web (Undertow)
- Webservices

2.6. WEB SERVER

Undertow バイトバッファプール

Undertow バイトバッファプールを使用してプールされた NIO **ByteBuffer** インスタンスを割り当てできるようになりました。すべてのリスナーにバイトバッファプールがあり、異なるバッファプールやワーカーを各リスナーに使用できます。バイトバッファプールはサーバーインスタンス間で共有することができます。

詳細は、『**Configuration Guide**』の「[Configuring Byte Buffer Pools](#)」を参照してください。

デフォルトのクッキーバージョンの設定

Undertow は、アプリケーションによって作成されたクッキーに使用するデフォルトのクッキーバージョンの設定方法を提供するようになりました。新しい **default-cookie-version** 属性に関する情報は、『**Configuration Guide**』の「[servlet-container Attributes](#)」を参照してください。

URL でエスケープ処理されていない文字を使用可能

Undertow を設定して、エスケープ処理されていない文字を URL で使用できるようになりました。これには、HTTP、HTTPS、および AJP リスナーに対して **allow-unescaped-characters-in-url** 属性を設定します。この属性が **true** に設定されていると、リスナーはエスケープ処理されていない ASCII でない文字が含まれる URL を処理します。**false** に設定されている場合、リスナーはエスケープ処理されていない ASCII でない文字が含まれる URL を **HTTP Bad Request 400** 応答コードで拒否します。

listener 属性に関する詳細は、『**Configuration Guide**』の「[Server Attributes](#)」を参照してください。

PROXY プロトコル

Undertow は、[The PROXY protocol Versions 1 & 2](#) 仕様で定義されている PROXY プロトコルバージョン 1 をサポートするようになりました。このオプションはデフォルトで無効になっており、同じプロトコルをサポートするロードバランサーの背後のリスナーのみに対して有効にする必要があります。これは、Undertow HTTP および HTTPS リスナー上の新しい **proxy-protocol** 属性を使用して設定されます。

listener 属性に関する詳細は、『**Configuration Guide**』の「[Server Attributes](#)」を参照してください。

Forwarded HTTP 拡張

JBoss EAP 7.2 には、[RFC 7239](#) を実装する **Forwarded** ハンドラーが導入され、リバースプロキシ背後のサーバーがヘッダー内のピアおよびローカルアドレスを受信できるようになりました。

通常、このハンドラーはリバースプロキシで有効になっている **X-Forwarded-*** ヘッダーと使用してはなりません。そのため、このハンドラーを使用するか、Undertow リスナーの **proxy-address-forwarding** 属性を有効にする必要があります。

セッションマネージャーの操作

`/deployment=DEPLOYMENT_NAME/subsystem=undertow` で管理 CLI から使用できるようになった、詳細なセッション情報を取得する操作は次のとおりです。

- **get-session-attribute**: セッションの特定の属性を返します。

- **get-session-creation-time**: セッション作成時間を ISO-8601 形式で取得します。
- **get-session-creation-time-millis**: UNIX エポックからセッション作成までの時間をミリ秒単位で取得します。
- **get-session-last-accessed-time**: セッションが最後にアクセスされた時間を ISO-8601 形式で取得します。
- **get-session-last-accessed-time-millis**: UNIX エポックからセッション最終アクセスまでの時間をミリ秒単位で取得します。
- **list-session-attribute-names**: session 属性名を表示します。
- **list-session-attributes**: セッションのすべての属性を表示します。
- **list-sessions**: アクティブなセッションをすべて表示します。

2.7. IO

新しい worker 属性

以前のリリースの JBoss EAP では、コアスレッドサイズは常に最大スレッドサイズと同じでした。そのため、**task-keepalive** 属性が設定された場合でもスレッドが停止しませんでした。本リリースでは、**task-core-threads** 属性を使用してコアスレッドプールのスレッド数を個別に設定できるようになったため、keepalive 設定が想定どおり動作するようになりました。

詳細は、JBoss EAP 『[Configuration Guide](#)』の「[Configuring a Worker](#)」および「[IO Subsystem Attributes](#)」を参照してください。

2.8. ロギング

ソケットログハンドラー

ソケットログハンドラーを設定して、ログメッセージを TCP または UDP ソケット上でリモートロギングサーバーへ送信できるようになりました。

詳細は、『[Configuration Guide](#)』の「[Configure a Socket Log Handler](#)」を参照してください。

JSON および XML フォーマッター

JSON および XML ログフォーマッターを使用してログメッセージを JSON または XML 形式に変換できます。

詳細は『[Configuration Guide](#)』の「[Log Formatters](#)」を参照してください。

2.9. トランザクション

新しい maximum-timeout トランザクションマネージャー属性

これまで、トランザクションタイムアウトを無制限のタイムアウトを意味する **0** に設定すると、トランザクションマネージャーはトランザクションタイムアウトの実際の値として **Integer.MAX_VALUE** を使用しました。最大整数値によって問題が発生することがあったため、トランザクションタイムアウト値の上限がより低くなりました。

新しい設定可能な属性である **maximum-timeout** が **transactions** サブシステムに追加されました。このデフォルト値は **31536000** 秒 (365日) です。トランザクションに無制限のタイムアウトを設定すると、トランザクションマネージャーは **maximum-timeout** の値を使用するようになり、**WARN** メッセージで動作がログに記録されることを通知するようになりました。

2.10. DATASOURCES

JDBC ドライバーのデータソースクラスプロパティの取得

datasource-class-info ランタイム属性は、JDBC ドライバーのデータソースクラスに設定できるデータソース接続プロパティのリストを提供します。管理コンソールを使用して XA データソースを追加または編集する場合や、非 XA データソースを編集する場合、**properties** フィールドはこのプロパティのリストを候補として提供します。



注記

JDBC ドライバーは、表示されるプロパティに対して設定された **driver-datasource-class-name** または **driver-xa-datasource-class-name** とともに作成されている必要があります。管理対象ドメインでプロパティを表示するには、JDBC ドライバーが含まれるプロファイルに、実行中のサーバーがある必要があります。

詳細は『[Configuration Guide](#)』の「[Datasource Attributes](#)」の表を参照してください。

2.11. EJB

HTTP/HTTPS 上の EJB および JNDI と HTTP ロードバランサー

リクエストが直接 HTTP リクエストにマップされるよう HTTP プロトコルを使用して EJB および JNDI 呼び出しを実行する機能が JBoss EAP 7.2 では完全サポートされるようになりました。また、EJB を HTTP ロードバランサー上で実行できます。詳細は『[Developing EJB Applications](#)』の「[EJB Invocation Over HTTP](#)」を参照してください。

コンテキストデータの EJB がクライアントへの返信

EJBClientInterceptor

は、**org.jboss.ejb.client.EJBClientInvocationContext#addReturnedContextDataKey (String key)** を呼び出してサーバー側の呼び出しコンテキストから特定のデータをリクエストすることができます。要求したデータがコンテキストデータマップの提供されるキー以下に存在する場合、クライアントに送信されます。

2.12. JSF

JSF デプロイメントでの DOCTYPE 宣言の拒否

管理 CLI を使用して、JSF デプロイメントの **DOCTYPE** 宣言を拒否することができます。

詳細は『[Configuration Guide](#)』の「[Disallowing DOCTYPE Declarations](#)」を参照してください。

2.13. HIBERNATE

Hibernate ORM 5.1 から Hibernate ORM 5.3 へのアップグレード

JBoss EAP 7.2 には Hibernate ORM 5.3 が含まれるようになりました。Hibernate ORM 5.3 には、Java 8 JDK を使用して構築され、実行時に Java 8 JRE を必要とした Hibernate ORM 5.2 の変更が含まれています。また、Hibernate ORM 5.3 には JPA 2.2 仕様のサポートも追加されています。この仕様に準拠するための変更やその他の改善内容が含まれています。

Hibernate ORM 5.2 および 5.3 で導入された機能に関する詳細と、アプリケーションを Hibernate ORM 5.1 から Hibernate ORM 5.3 に移行するために必要な事柄については、JBoss EAP『[移行ガイド](#)』の「[Hibernate ORM 5.1 から Hibernate ORM 5.3 への移行](#)」を参照してください。

Hibernate Validator 5.3.x から Hibernate Validator 6.0.x へのアップグレード

JBoss EAP 7.2 には、[JSR 380: Bean Validation 2.0](#) のリファレンス実装である Hibernate Validator 6.0.x が含まれています。

詳細は、JBoss EAP 『開発ガイド』の「[Bean Validation](#)」を参照してください。

2.14. クラスタリング

改良された **execute** メソッドによって使用される **CompletableFuture**

本リリースでは、新しい Java EE 8 の **CompletableFuture** インターフェースを利用するために **CommandDispatcher** 非同期メソッドが改良されました。これにより、**CommandDispatcher** のコンシューマーはディスパッチしたコマンドの非ブロッキング処理を実装できます。

非推奨となったメソッド	新たな推奨メソッド
executeOnNode	executeOnMember
executeOnCluster	executeOnGroup
submitOnNode	executeOnMember
submitOnCluster	executeOnGroup

詳細は、『開発ガイド』の「[クラスタリングサービスのパブリック API](#)」を参照してください。

2.15. INFINISPAN

HotRod クライアントインジェクション

@Resource JNDI インジェクションを使用して、HotRod クライアントをインジェクトし、リモート JDG クラスタに接続することができます。

詳細は、『**Configuration Guide**』の「[Externalize HTTP Sessions to JBoss Data Grid](#)」を参照してください。

非ブロッキングの初期状態遷移

状態遷移が完了するまで待たなくても、キャッシュが即座に利用可能になります。これは、キャッシュの **timeout** 属性を **0** に設定し、バックグラウンド操作にてキャッシュの状態を受信できるようにすると実現できます。

詳細は『**Configuration Guide**』の「[State Transfer](#)」を参照してください。

散在 (**scattered**) キャッシュモード

infinispan サブシステムは散在 (scattered) キャッシュモードをサポートするようになりました。散在 (scattered) モードは、一貫したハッシュアルゴリズムを使用して所有者を判断する分散モードと似ています。しかし、所有者は 2 人のメンバーに限定され、オリジネーター (指定のセッションのリクエストを受信するノード) は常にロックを調整する所有者やキャッシュエントリのアップデートを想定します。散在 (scattered) モードで使用されるキャッシュ書き込みアルゴリズムは、書き込み操作の結果が単一の RPC 呼び出しになることを保証します。これにより、競合の可能性を下げ、クラスタポートロギーの変更後にパフォーマンスを向上することが可能になります。

詳細は『**Configuration Guide**』の「[Clustering Modes](#)」を参照してください。

リモートキャッシュストアを使用した **HTTP** セッションの外部化

HTTP セッションを JBoss Data Grid に外部化する新しいメソッドが本リリースに含まれています。このメソッドは、セキュリティのためにクライアント SSL コンテキストが定義されている JBoss EAP の **infinispan** サブシステムにあるリモートキャッシュコンテナを利用します。

管理 CLI および管理コンソールからリモートキャッシュコンテナを設定できます。

詳細は、『**Configuration Guide**』の「[Externalize HTTP Sessions to JBoss Data Grid](#)」を参照してください。

2.16. WEB サービス

JSON バインディングの Java API

RESTEasy は JSON-B と JSON-P の両方をサポートします。仕様によると、**JsonValue** とそのサブタイプを除き、すべてのタイプのエンティティで、JSON-B のエンティティプロバイダーの優先度は JSON-P のエンティティプロバイダーよりも高くなります。

resteasy-json-binding-provider モジュールからの **JsonBindingProvider** は、JSON-B のサポートを提供します。JAX-RS 2.1 の要件を満たすため、**JsonBindingProvider** プロバイダーの優先度は、Jackson ペイロードなどの JSON ペイロードを処理する他のプロバイダーよりも高くなります。後方互換性を維持するには、**resteasy.preferJacksonOverJsonB** コンテキストプロパティを **true** に設定し、現在のデプロイメントの **JsonBindingProvider** 設定を無効にします。

詳細は、JBoss EAP 『**Developing Web Services Applications**』の「[Java API for JSON Binding](#)」を参照してください。

非同期 HTTP リクエストの処理

RESTEasy のデフォルトの非同期エンジン実装クラスは **ApacheHttpAsyncClient4Engine** です。**ResteasyClientBuilder** クラスで **useAsyncHttpEngine** メソッドを呼び出すと、非同期エンジンをアクティブなエンジンとして設定できます。

詳細は、JBoss EAP 『**Developing Web Services Applications**』の「[Asynchronous NIO Request Processing](#)」を参照してください。

カスタム RESTEasy アノテーション

バイトコードにパラメーター名が追加されたた

め、**@PathParam**、**@QueryParam**、**@FormParam**、**@CookieParam**、**@HeaderParam**、および **@MatrixParam** アノテーションでパラメーター名を指定する必要がなくなりました。これには、任意の value パラメーターを持つ異なるパッケージで、同じ名前を持つ新しいアノテーションに切り替える必要があります。

詳細は、JBoss EAP 『**Developing Web Services Applications**』の「[Custom RESTEasy Annotations](#)」を参照してください。

ParamConverter 機能の拡張

JAX-RS セマンティックでは、**ParamConverter** は各オブジェクトを表す単一の文字列を変換します。RESTEasy はセマンティックを拡張して、**ParamConverter** が複数オブジェクトの文字列表現を解析できるようにし、**List<T>**、**Set<T>**、**SortedSet<T>**、アレイ、またはその他の複数値のデータ構造の生成を可能にします。

詳細は、JBoss EAP 『**Developing Web Services Applications**』の「[Extending the Functionality of the ParamConverter](#)」を参照してください。

リソースメソッドアルゴリズムの切り替え

3.0.25.Final より前の RESTEasy 3.0.x バージョンで使用されたリソースメソッド一致アルゴリズムでバグが見つかりました。このバグにより、リクエストの応答時に RESTEasy は余分なリソースメソッド

ドを返しました。詳細は『[移行ガイド](#)』の「[JAX-RS および RESTEasy のアプリケーションの変更](#)」を参照してください。

RESTEasy サービスプロバイダーインターフェース

JBoss EAP は、**ResourceBuilder** を使用して作成されるリソースクラスメタデータを編集するための RESTEasy サービスプロバイダーインターフェース (SPI) を提供するようになりました。**ResourceClassProcessor** インターフェースの実装によって、メタデータ生成のカスタマイズが可能になります。

RESTEasy SPI に関する詳細は、JBoss EAP 『[Developing Web Services Applications](#)』の「[RESTEasy SPI to Modify Resource Metadata](#)」を参照してください。

HTTP リダイレクトの JAX-RS クライアントサポート

Apache **HttpClient** をベースとする JAX-RS **ClientHttpEngine** 実装は HTTP リダイレクトをサポートします。詳細は『[Developing Web Services Applications](#)』の「[HTTP Redirect](#)」を参照してください。

2.17. メッセージング

IBM MQ リソースアダプター

本リリースの JBoss EAP は以下の構成でテストされています。

- IBM MQ 8.0.0.10 リソースアダプターは IBM MQ 8.0.0.x ブローカーに対してテストされました。バージョンが 8.0.0.0 から 8.0.0.9 までの IBM MQ リソースアダプターはサポートされません。
- IBM MQ 9.0.0.4 リソースアダプターは IBM MQ 9.0.0.x ブローカーに対してテストされました。バージョンが 9.0.0.0 から 9.0.0.3 までの IBM MQ リソースアダプターはサポートされません。

IBM MQ リソースアダプターに関する詳細は、JBoss EAP 『[Configuring Messaging](#)』の「[Deploying the IBM MQ Resource Adapter](#)」を参照してください。

JDBC データベースを使用したメッセージングジャーナルの永続化

本リリースの JBoss EAP では、現在サポートされている Oracle 12c データベースの他に、JDBC を使用してメッセージを永続化する場合に IBM DB2 Enterprise データベースをサポートします。

メッセージング JDBC 永続ストアの HA トポロジのサポート

本リリースの JBoss EAP では、メッセージング JDBC 永続ストアの HA トポロジをサポートします。詳細は、JBoss EAP 『[Configuring Messaging](#)』の「[Configuring HA for Messaging JDBC Persistence Store](#)」を参照してください。

リモート Red Hat AMQ 7 メッセージングブローカーへの接続の簡略化

リモート Red Hat AMQ 7 メッセージングブローカーへ接続するのに、JBoss EAP の埋め込みメッセージングブローカーが必要なくなりました。リモート Red Hat AMQ ブローカーに直接接続するために必要なリソースは **messaging-activemq** サブシステムで定義できます。

統合された Artemis リソースアダプターを使用した Red Hat AMQ への接続

統合された Artemis リソースアダプターを設定し、JBoss EAP 7.2 アプリケーションの JMS プロバイダーとなる Red Hat AMQ 7 のリモートインストールへ接続することができます。これにより、JBoss EAP がリモート Red Hat AMQ 7 サーバーのクライアントになることができます。

詳細は、JBoss EAP 『[Configuring Messaging](#)』の「[Configuring the Artemis Resource Adapter to Connect to Red Hat JBoss AMQ 7](#)」を参照してください。

2.18. OPENSIFT

JBoss EAP でネイティブに統合された KUBE_PING

これまで、**KUBE_PING** JGroups ディスカバリープロトコルは JBoss EAP OpenShift イメージにのみ実装されました。**KUBE_PING** は JBoss EAP にネイティブ実装されるようになったため、独自のカスタムコンテナイメージを作成するユーザーはクラスター化されたアプリケーションでネイティブの **KUBE_PING** を使用できるようになりました。**KUBE_PING** の使用に関する詳細は、『[Getting Started with JBoss EAP for OpenShift Container Platform](#)』の「[Clustering reference](#)」を参照してください。

2.19. モジュール

事前定義されたモジュール

すべての JBoss モジュール API が含まれる `org.jboss.modules` は事前定義されたモジュールのセットで、デフォルトのモジュールローダーを使用する場合に JBoss EAP 7.2 でサポートされます。この特別なモジュールは常に利用可能で、JBoss Modules によって提供されます。Java 9 以上で提供される標準の Java Platform Module System (JPMS) モジュールも標準の名前で利用可能です。JDK 8 を使用する場合、JDK 9 モジュールは JBoss Module によってエミュレートされます。

詳細は『[Configuration Guide](#)』の「[Predefined Modules](#)」を参照してください。

2.20. クイックスタートおよび BOM

アプリケーションの開発に使用可能な JBoss EAP BOM

Java EE 8 のアップデートに伴い、JBoss EAP Maven BOM ファイルのアーティファクト ID が変更になりました。以下の表は、本リリースでアプリケーションの開発に使用できる Maven BOM を表しています。

BOM アーティファクト ID	ユースケース
<code>jboss-eap-javaee8</code>	JBoss EAP Java EE 8 API および追加の JBoss EAP API JAR をサポート。
<code>jboss-eap-javaee8-with-spring4</code>	<code>jboss-eap-javaee8</code> および推奨される Spring 4 バージョン。
<code>jboss-eap-javaee8-with-tools</code>	<code>jboss-eap-javaee8</code> および Arquillian などの開発ツール。

アプリケーション開発で使用できる BOM に関する詳細は、『[開発ガイド](#)』の「[プロジェクト依存関係の管理](#)」を参照してください。

第3章 テクノロジーレビュー



重要

以下の設定および機能は、テクノロジーレビューとしてのみ提供されます。テクノロジーレビューの機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあるため、Red Hat は本番環境での使用は推奨しません。テクノロジーレビューの機能は、最新の技術をいち早く提供して、開発段階で機能のテストやフィードバックの収集を可能にするために提供されま

す。

テクノロジーレビュー機能のサポート範囲については、Red Hat カスタマーポータル の「[テクノロジーレビュー機能のサポート範囲](#)」を参照してください。

新しい **Agroal** データソースサブシステム

JBoss EAP 7.2 には **datasources-agroal** サブシステムが導入されました。これは、プールでの競合が深刻な状態で非常に優れたパフォーマンスを発揮するライトウェイトなデータソース接続プール実装です。現在の JCA ベースの **datasources** サブシステムの代わりに使用することができます。

このサブシステムは、デフォルトの JBoss EAP 設定では有効になっていません。Agroal データソースの有効化および使用に関する詳細は、『[Configuration Guide](#)』の「[Datasource Management With Agroal](#)」を参照してください。

MicroProfile REST クライアント

JBoss EAP は JAX-RS 2.0 クライアント上に構築される MicroProfile REST クライアントをサポートするようになりました。これは、タイプセーフな方法で HTTP 上で RESTful サービスを呼び出します。MicroProfile REST クライアントを使用すると、実行可能なコードでクライアントアプリケーションを作成できます。Java インターフェースとして定義される MicroProfile TypeSafe REST クライアントは、プロバイダーのプログラムによる登録および宣言的な登録を有効にします。

MicroProfile REST クライアントに関する詳細は、JBoss 『[Developing Web Services Applications](#)』の「[MicroProfile Rest Client](#)」を参照してください。

非同期リクエスト処理および反応的戻り値の型の **RESTEasy** サポートの拡張

JBoss EAP は非同期リクエスト処理と反応的な戻り値の型の RESTEasy サポートを拡張しました。これには、プラグ可能な反応的な型や、追加の反応的クラスの拡張のサポートが含まれます。**RxInvoker** という新型のインボーカーと、**CompletionStageRxInvoker** というこの型のデフォルト実装もサポートされます。JBoss EAP は、フィルターを非同期フィルターに変換する機能も提供します。直感的なプログラミングをサポートする RESTEasy 拡張であるプロキシは、**CompletionStage** と RxJava2 タイプ (**Single**、**Observable**、および **Flowable**) の両方を含めるために拡張されました。

詳細は、JBoss EAP 『[Developing Web Services Applications](#)』の「[Extending RESTEasy Support for Asynchronous Request Processing and Reactive Return Types](#)」を参照してください。

Eclipse MicroProfile Config

JBoss EAP 7.2 は、**microprofile-config-smallrye** サブシステムを使用して **Eclipse MicroProfile Config** を提供する **SmallRye Config** コンポーネントをサポートするようになりました。これにより、変更や再パッケージの必要なく、アプリケーションやマイクロサービスが複数の環境で実行されるように設定することが可能になります。

詳細は『[Configuration Guide](#)』の「[Using Eclipse MicroProfile Config to Manage Configuration](#)」を参照してください。

Eclipse MicroProfile OpenTracing

JBoss EAP 7.2 は、[Eclipse MicroProfile OpenTracing](#) 仕様のサポートを提供する [SmallRye OpenTracing](#) コンポーネントを実装します。これは、JBoss EAP サーバーにデプロイされたアプリケーションやサーバーを確認しながらリクエストの追跡を可能にし、リクエストのライフサイクルの監視機能を拡張します。

`microprofile-opentracing-smallrye` サブシステムに関する詳細は、『[Configuration Guide](#)』の「[Tracing Requests with the MicroProfile OpenTracing SmallRye Subsystem](#)」を参照してください。CDI Bean および JAX-RS エンドポイントの追跡をカスタマイズする方法については、『[開発ガイド](#)』の「[Eclipse MicroProfile OpenTracing を使用したリクエストのトレース](#)」を参照してください。

Eclipse MicroProfile Health

JBoss EAP 7.2 には、[Eclipse MicroProfile Health](#) 機能を提供する [SmallRye Health](#) コンポーネントが含まれます。この機能を使用すると、リモートノードのヘルス状態をチェックでき、JBoss EAP インスタンスが想定どおりに応答しているかどうかを判断することができます。`microprofile-health-smallrye` サブシステムはデフォルトの JBoss EAP 7.2 設定に含まれています。

`microprofile-health-smallrye` サブシステムに関する詳細は、『[Configuration Guide](#)』の「[Monitor Server Health Using the MicroProfile Health Check](#)」を参照してください。カスタムヘルスチェックの実装方法に関する詳細は、『[開発ガイド](#)』の「[カスタムヘルスチェックの実装](#)」を参照してください。

第4章 サポートされない機能および非推奨の機能

4.1. サポートされない機能

メンテナンスコストの高さ、コミュニティでの関心の低さ、および高性能な代替ソリューションの存在などが理由で、一部の技術がサポート対象外になりました。以下の機能は JBoss EAP 7.2 ではサポートされません。



注記

本リリースノートの「[新機能および改良された機能](#)」に記載されている場合の除き、『7.1.0 リリースノート』の「[サポートされない機能](#)」に記載されている機能は JBoss EAP 7.2 でもサポートされません。

プラットフォームおよび機能

本リリースでは、以下のプラットフォームおよび機能のサポートが除外されました。

オペレーティングシステムおよび関連する web サーバー

- Windows Server 2008 および関連する IIS Web サーバー
- Solaris 10 および関連する web サーバー
- x86_64 上の Solaris 11 および関連する web サーバー
- HP-UX
- Red Hat Enterprise Linux 6 32-bit

Java 仮想マシン

- HP-UX

データベースおよびデータベースコネクタ

- IBM DB2 e9.7 および e10.5
- MySQL 5.5
- Microsoft SQL Server 2012 SP3
- Microsoft SQL Server 2014 SP2
- PostgreSQL 9.3、9.4、および 9.6
- EnterpriseDB 9.3、9.4、および 9.6
- Sybase ASE 15.7
- Oracle 11g R2 および 11g R2 RAC

JMS プロバイダーおよびアダプター

- IBM WebSphere MQ 7.5

LDAP サーバー

- Red Hat Directory Server 9.1
- Microsoft Active Directory 2008

テスト済みのフレームワーク

- JQuery (全バージョン)
- AngularJS (全バージョン)

メッセージング (ActiveMQ Artemis)

異なるバージョンの JBoss EAP で構成されるクラスターでは、JGroups を使用したディスカバリーグループやブロードキャストグループの設定はサポートされません。詳細は、『[Configuring Messaging](#)』の「[Clusters Overview](#)」および『[パッチおよびアップグレードガイド](#)』の「[クラスターのアップグレード](#)」を参照してください。

RPM current リポジトリ

本リリースでは、JBoss EAP **current** リポジトリは利用できません。JBoss EAP **current** リポジトリにサブスクライブしている場合、サブスクリプションを JBoss EAP の **minor** リポジトリの 1 つに変更する必要があります。詳細は、『[インストールガイド](#)』の「[リポジトリの変更](#)」を参照してください。[RPM インストールのパッチ](#) および [RPM インストールのアップグレード](#) に関する情報は、『[パッチおよびアップグレードガイド](#)』を参照してください。

クイックスタート

JBoss EAP 7.1 で利用可能な以下のクイックスタートは本リリースでは使用できません。

- cdi-alternative
- cdi-decorator
- cdi-injection
- cdi-interceptors
- cdi-portable-extension
- cdi-stereotyp
- cdi-veto
- ejb-security-interceptors
- forge-from-scratch
- h2-console
- kitchensink-html5-mobile
- kitchensink-ml-ear
- log4j
- picketlink-sts
- shrinkwrap-resolver

- spring-kitchensink-asyncrequestmapping
- spring-kitchensink-controlleradvice
- spring-kitchensink-matrixvariables
- spring-petclinic
- tasks
- xml-dom4j

4.2. 非推奨の機能

本リリースでは一部の機能が非推奨になりました。非推奨の機能には改良が加えられず、将来的 (通常は次のメジャーリリース) に削除される可能性があります。

Red Hat は標準のサポート条件に基づき、継続して完全サポートおよびバグ修正を提供します。Red Hat のサポートポリシーに関する詳細は、Red Hat カスタマーポータルに記載されている [Red Hat JBoss Middleware の製品アップデートとサポートポリシー](#) を参照してください。

非推奨となった機能の詳細は、Red Hat カスタマーポータルに記載されている [JBoss Enterprise Application Platform コンポーネントの詳細](#) を参照してください。

IO サブシステム

- IO バッファプールは本リリースで非推奨となりました。これは Undertow バイトバッファプールに置き換えられます。

キャッシュストア

- **hotrod** キャッシュストアの使用が推奨されるため、**remote** キャッシュストアは非推奨となりました。

Red Hat JBoss Operations Network

JBoss EAP を管理するための Red Hat JBoss Operations Network の使用は、本リリースで非推奨となりました。

プラットフォームおよび機能

本リリースでは、以下のプラットフォームおよび機能のサポートが非推奨となりました。

オペレーティングシステムおよび関連する web サーバー

- Windows Server 2012 R2 および関連する IIS web サーバー
- SPARC 上の Solaris 11 および関連する web サーバー

JMS プロバイダーおよびアダプター

- TIBCO EMS

LDAP サーバー

- Microsoft Active Directory 2012 R2

テスト済みのフレームワーク

- Spring (Core)、Spring Security、Spring Web Flow、Spring Web Service

クラウド

- Amazon EC2
- Microsoft Azure

ドキュメント

JBoss EAP の以下のガイドは非推奨となりました。

- [Deploying JBoss EAP on Amazon Web Services](#)
- [Using JBoss EAP in Microsoft Azure](#)

第5章 解決した問題

本リリースで解決した問題のリストは、[Resolved Issues for JBoss EAP 7.2 GA](#) を参照してください。

第6章 修正された CVE

JBoss EAP 7.2 には、以下のセキュリティー関連の問題修正が含まれています。

- **CVE-2017-7503: xml frameworks:** JBoss EAP 7.0.5 implementation of `javax.xml.transform.TransformerFactory` is vulnerable to XXE (**xml frameworks: javax.xml.transform.TransformerFactory** の JBoss EAP 7.0.5 実装は XXE に対して脆弱である)
- **CVE-2018-10237: guava:** Unbounded memory allocation in `AtomicDoubleArray` and `CompoundOrdering` classes allow remote attackers to cause a denial of service (**guava: AtomicDoubleArray** および `CompoundOrdering` クラスのバインドされていないメモリ割り当てによってリモート攻撃者によるサービス拒否攻撃が可能)
- **CVE-2018-1067: undertow:** HTTP header injection using CRLF with UTF-8 encoding (**undertow:** UTF-8 エンコードでの CRLF を使用した HTTP ヘッダーのインジェクト)
- **CVE-2018-10862: wildfly-core:** Path traversal can allow the extraction of `.war` archives to write arbitrary files (**wildfly-core:** パストラバーサルによって `.war` アーカイブの抽出で任意のファイルに書き込みが可能)
- **CVE-2017-12174: artemis/hornetq:** Memory exhaustion via UDP and JGroups discovery (UDP および JGroups の検索によるメモリーの枯渇)
- **CVE-2017-12629: Solr:** Code execution via entity expansion (**Solr:** エンティティ拡大によるコード実行)
- **CVE-2017-15089: infinispan:** Unsafe deserialization of malicious object injected into data cache (**infinispan:** データキャッシュにインジェクトされた悪意のあるオブジェクトの安全でないデシリアライズ)
- **CVE-2017-12196: undertow:** Client can use bogus uri in Digest authentication (**undertow:** クライアントは Digest 認証で偽の URI を使用できる)
- **CVE-2018-8088: slf4j:** Deserialisation vulnerability in `EventData` constructor can allow for arbitrary code execution (**slf4j:** `EventData` コンストラクターのデシリアライズの脆弱性によって任意コードの実行が可能になる)
- **CVE-2018-1047: undertow:** Path traversal in `ServletResourceManager` class (**undertow:** `ServletResourceManager` クラスのパストラバーサル)
- **CVE-2018-8039: apache-cxf:** TLS hostname verification does not work correctly with `com.sun.net.ssl.*` (**apache-cxf: com.sun.net.ssl.*** を用いた TLS ホスト名の検証が適切に動作しない)

第7章 既知の問題

本リリースでの既知の問題の一覧は、[Known Issues for JBoss EAP 7.2 GA](#) を参照してください。

この他に、以下に注意してください。

- JDK 11 の `jboss-cli-client.jar` ファイルを使用して開始された管理 CLI インスタンスから埋め込みサーバーの起動を試みると、エラー **WFLYEMB0014: Cannot load module** が発生します。このエラーが発生しないようにするには、管理 CLI の起動時に `--add-modules java.se` を追加する必要があります。

```
$ java --add-modules java.se -jar jboss-cli-client.jar
```

- JBoss EAP のデフォルトでは、重複した XA リソースが重複した `xa_end` 呼び出しの **XAER_PROTO** を返すことができます。これは、JTA 1.2 で想定される各リソースが `xa_end` 呼び出しを実行する動作と対照的です。これは、[JBEAP-12671](#) に説明があるように、JBoss EAP には現在 JTA 1.2 の重複した `xa_end` 呼び出しの動作をサポートしない Artemis が同梱されているからです。この厳密な動作を完全に JTA 1.2 に準拠させるには、フラグ `DJTAEnvironmentBean.strictJTA12DuplicateXAENDPROTOErr=true` を渡します。しかし、Artemis の使用時にはエラーが発生し、重複した Artemis リソースが含まれるトランザクションをコミットできません。
- クライアントアプリケーションが直接 Artemis クライアント JAR (`artemis-jms-client`、`artemis-commons`、`artemis-core-client`、`artemis-selector` など) に依存する場合、`wildfly-client-properties` の `pom.xml` ファイルに依存関係を追加する必要があります。

```
<dependency>  
  <groupId>org.jboss.eap</groupId>  
  <artifactId>wildfly-client-properties</artifactId>  
</dependency>
```

これにより、[JBEAP-15889](#) に記載されているように、旧バージョンの JBoss EAP 7 クライアントから `message.getJMSReplyTo()` を呼び出すときに `JMSRuntimeException` が発生しないようにします。

- JBoss EAP 7.2 RPM パッケージを RHEL 7 上にインストールし、デフォルトの RHEL JDK 11 パッケージのみが含まれる場合、現在パッケージに定義された依存関係が原因で、RPM は JDK 1.8 もインストールします。その結果、JDK 1.8 がデフォルトの JDK として設定されるため、`alternatives` を使用してシステムが JDK 11 を使用するよう設定する必要があります。

```
alternatives --config java
```

JDK 11 に対応する番号を入力します。

