



## Red Hat Data Grid 7.3

### Red Hat Data Grid 7.3 リリースノート

Data Grid のリリース情報



Data Grid のリリース情報

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

The OpenStack<sup>®</sup> 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

Red Hat Data Grid は、高パフォーマンスの分散インメモリーデータストアです。これらのリリースノートは、Data Grid 7.3.x の累積的なリリース情報を提供します。Data Grid 7.3 の各バージョンのすべての機能と機能強化を確認し、現在の既知の問題と解決済みの問題に関する情報を見つけ、Red Hat がサポートする設定と Data Grid 7.3 のコンポーネントバージョンを確認できます。

## 目次

パート I. DATA GRID 7.3 の新機能 .....	5
第1章 7.3.11 の更新と機能拡張 .....	6
1.1. DATA GRID 7.3.11 のセキュリティー更新 .....	6
第2章 7.3.10 の更新と機能強化 .....	7
2.1. DATA GRID 7.3.10 のセキュリティー更新 .....	7
第3章 7.3.9 の更新と機能強化 .....	8
3.1. LOG4J バージョン 1.X の削除 .....	8
3.2. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.10 へのアップグレード .....	8
3.3. JBOSS マーシャリングのアップグレード .....	8
第4章 7.3.8 の更新と機能強化 .....	9
4.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.4 へのアップグレード .....	9
第5章 7.3.7 の更新と機能強化 .....	10
5.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.9 へのアップグレード .....	10
5.2. ローリングアップグレードパフォーマンスの向上 .....	10
第6章 7.3.6 の更新と機能強化 .....	11
6.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.7 へのアップグレード .....	11
6.2. SPRING バージョンのアップグレード .....	11
6.3. RED HAT DATA GRID エンドポイントを開始および停止する機能 .....	11
6.4. HOT ROD アクセスログにタスク名が含まれる .....	11
6.5. クロスサイトレプリケーションパフォーマンスの複数の向上 .....	11
第7章 7.3.5 の更新と機能強化 .....	12
7.1. クラスター化された最大アイドル有効期限 .....	12
7.2. ソケットタイムアウトの例外による接続の詳細の提供 .....	12
7.3. HOT ROD クライアント接続プールの設定 .....	12
7.4. DATA GRID のドキュメント .....	12
第8章 7.3.4 の更新と機能強化 .....	14
8.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.5 へのアップグレード .....	14
8.2. SPRING BOOT バージョンのアップグレード .....	14
8.3. クラスター化されたキャッシュを持つプライマリ所有者からの期限切れ .....	14
8.4. クロスサイトレプリケーションの有効期限コマンド .....	14
8.5. RED HAT DATA GRID CLI セッションの有効期限 .....	14
第9章 7.3.3 の更新と機能強化 .....	15
9.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.4 へのアップグレード .....	15
9.2. AMAZON WEB SERVICES (AWS) のネイティブ S3 PING .....	15
9.3. JGROUPS トランスポートスタックでのフロー制御の変更 .....	15
9.4. REST エンドポイント承認 .....	15
9.5. PROTOSTREAM ライブラリーの 4.2.4.FINAL への更新 .....	16
第10章 7.3.2 の更新と機能強化 .....	17
10.1. 非同期クロスサイトレプリケーションでサイトを自動的にオフラインにする .....	17

10.2. JMX による非同期クロスサイトレプリケーション統計	17
10.3. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.1 との依存関係の調整	17
10.4. リスナー通知のスキップ	17
<b>第11章 7.3.1 の更新と機能強化</b>	<b>18</b>
11.1. JAVA 11	18
11.2. OPENJDK 8 FOR MICROSOFT WINDOWS	18
11.3. JGROUPS の 4.0.18.FINAL への更新	18
11.4. JGROUPS DNS_PING	18
11.5. DATA GRID FOR OPENSIFT の設定可能なログレベル	18
11.6. ファイル名アナライザー	18
<b>第12章 7.3.0 の新機能と機能強化</b>	<b>19</b>
12.1. DATA GRID FOR OPENSIFT の向上	19
12.2. フレームワークの統合	19
12.3. HOT ROD クライアントの向上	21
12.4. 永続性の向上	23
12.5. データの相互運用性	24
12.6. デフォルトのアナライザー	24
12.7. メトリックの向上	24
12.8. ストリームのロック	25
12.9. 設定の向上	25
12.10. キャッシュ操作のセキュリティーの向上	25
12.11. HTTP/2 のサポート	25
<b>第13章 DATA GRID 7.3 のテクノロジープレビュー</b>	<b>26</b>
13.1. RED HAT OPENSIFT でのクロスサイトレプリケーション	26
13.2. C++ および C# クライアントによるリモート管理	27
13.3. C++ および C# クライアントを使用したクラスター化されたカウンター	27
13.4. 管理コンソールの機能	27
<b>第14章 DATA GRID 7.3 で廃止された特徴と機能</b>	<b>28</b>
<b>パート II. サポートされる構成とコンポーネントのバージョン</b>	<b>30</b>
<b>第15章 DATA GRID 7.3 でサポートされる構成</b>	<b>31</b>
<b>第16章 DATA GRID コンポーネントのバージョン</b>	<b>32</b>
<b>パート III. 既知の問題および修正された問題</b>	<b>33</b>
<b>第17章 DATA GRID 7.3 の既知の問題</b>	<b>34</b>
<b>第18章 修正された問題</b>	<b>38</b>
18.1. DATA GRID 7.3.9 で修正済み	38
18.2. DATA GRID 7.3.8 で修正済み	38
18.3. DATA GRID 7.3.7 で修正済み	38
18.4. DATA GRID 7.3.6 で修正済み	38
18.5. DATA GRID 7.3.5 で修正済み	39
18.6. DATA GRID 7.3.4 で修正済み	39
18.7. DATA GRID 7.3.3 で修正済み	40
18.8. DATA GRID 7.3.2 で修正済み	40
18.9. DATA GRID 7.3.1 で修正済み	41
18.10. DATA GRID 7.3.0 で修正済み	41
<b>パート IV. DATA GRID 7.3 への移行</b>	<b>42</b>

---

<b>第19章 DATA GRID 7.3 の変更点</b> .....	<b>43</b>
19.1. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP)	43
19.2. EAP モジュール	43
19.3. JBOSS FUSE の CAMEL コンポーネント	43
19.4. キャッシュストアの互換性	43
19.5. MEMCACHED ストレージ	43
19.6. MEMCACHED コネクター	44
19.7. スクリプトの応答	44
19.8. サーバースレッドプール	44
19.9. パッシベーションを使用した共有キャッシュストア	44
19.10. AFFINITYINDEXMANAGER を使用したデフォルトのシャード	44
19.11. ADVANCEDCACHELOADER の変更	44
19.12. DATA GRID 7.3 での非推奨事項	44
<b>パート V. DATA GRID サーバーへのパッチ適用</b> .....	<b>45</b>
<b>第20章 RED HAT DATA GRID へのパッチの適用</b> .....	<b>46</b>
7.3.8 および 7.3.9 パッチによるエラーの解決	46
<b>第21章 パッチを元に戻す</b> .....	<b>48</b>





## パート I. DATA GRID 7.3 の新機能

## 第1章 7.3.11 の更新と機能拡張

### 1.1. DATA GRID 7.3.11 のセキュリティー更新

Data Grid 7.3.11 は、CVE に対処するためのセキュリティー強化を提供します。Data Grid 7.3 デプロイメントは、可能なかぎり早急にバージョン 7.3.11 にアップグレードする必要があります。詳細は、このリリースに関連するアドバイザリー [RHSA-2023:6286](#) を参照してください。

Red Hat は、デプロイメントを 7.3.x から最新の Data Grid 8 バージョンに可能なかぎり早急にアップグレードすることを推奨します。Data Grid チームは定期的にセキュリティーの脆弱性にパッチを適用し、ソフトウェアの最新バージョンの問題を積極的に修正します。

最新の Data Grid ドキュメントは、[Data Grid 製品ドキュメント](#) にあります。

Data Grid のバージョンライフサイクルとサポートの詳細については、[Data Grid 製品の更新とサポートポリシー](#) を参照してください。

## 第2章 7.3.10 の更新と機能強化

### 2.1. DATA GRID 7.3.10 のセキュリティー更新

Data Grid 7.3.10 は、CVE に対処するためのセキュリティー強化を提供します。Data Grid 7.3 デプロイメントは、可能なかぎり早急にバージョン 7.3.10 にアップグレードする必要があります。詳細については、このリリース [RHSA-2023:1303](#) に関連するアドバイザリーを参照してください。

Red Hat は、デプロイメントを 7.3.x から最新の Data Grid 8 バージョンに可能なかぎり早急にアップグレードすることを推奨します。Data Grid チームは定期的にセキュリティーの脆弱性にパッチを適用し、ソフトウェアの最新バージョンの問題を積極的に修正します。

最新の Data Grid ドキュメントは、[Data Grid 製品ドキュメント](#) にあります。

Data Grid のバージョンライフサイクルとサポートの詳細については、[Data Grid 製品の更新とサポートポリシー](#) を参照してください。

## 第3章 7.3.9 の更新と機能強化

### 3.1. LOG4J バージョン 1.X の削除

Data Grid 7.3.9 は、Log4j バージョン 1.x コンポーネントを削除して、重大なセキュリティの脆弱性と中程度の重大度の他の CVE に対処します。Data Grid 7.3 デプロイメントは、可能なかぎり早急にバージョン 7.3.9 にアップグレードする必要があります。

Data Grid 7.3.9 には Log4j バージョン 1.x が同梱されていないため、キャッシュローダーやサーバータスクの実装など、Log4j バージョン 1.x の機能に依存するカスタムコンポーネントは機能しなくなりました。

Red Hat は、デプロイメントを 7.3.x から最新の Data Grid 8 バージョンに可能なかぎり早急にアップグレードすることを推奨します。Data Grid チームは定期的にセキュリティの脆弱性にパッチを適用し、ソフトウェアの最新バージョンの問題を積極的に修正します。

最新の Data Grid ドキュメントは、[Data Grid 製品ドキュメント](#) にあります。

Data Grid のバージョンライフサイクルとサポートの詳細については、[Data Grid 製品の更新とサポートポリシー](#) を参照してください。

### 3.2. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.10 へのアップグレード

Data Grid サーバーは EAP 7.3.10 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。



#### 重要

Data Grid サーバーのインストールにパッチを適用して、7.3.9 にアップグレードすると、パッチの問題により、起動時に、エラーが発生します。パッチを適用した後、設定を手動で編集する必要があります。

手順については、[7.3.8 および 7.3.9 パッチによるエラーの解決](#) を参照してください。

### 3.3. JBOSS マーシャリングのアップグレード

Data Grid 7.3.9 は、JBoss マーシャリングをバージョン 2.0.12.Final にアップグレードします。

## 第4章 7.3.8 の更新と機能強化

### 4.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.4 へのアップグレード

Data Grid サーバーは EAP 7.3.4 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。



#### 重要

Data Grid サーバーのインストールにパッチを適用して、7.3.8 にアップグレードすると、パッチの問題により、起動時に、エラーが発生します。パッチを適用した後、設定を手動で編集する必要があります。

手順については、[7.3.8 および 7.3.9 パッチによるエラーの解決](#) を参照してください。

## 第5章 7.3.7 の更新と機能強化

### 5.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.9 へのアップグレード

Data Grid サーバーは EAP 7.2.9 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。

### 5.2. ローリングアップグレードパフォーマンスの向上

Data Grid クラスターのローリングアップグレードを実行すると、ターゲットクラスターはソースクラスターからより効率的にデータを取得します。

## 第6章 7.3.6 の更新と機能強化

### 6.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.7 へのアップグレード

Data Grid サーバーは EAP 7.2.7 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。

### 6.2. SPRING バージョンのアップグレード

Data Grid のこのリリースは、次のように Spring バージョンにアップグレードされます。

- Spring Boot は **2.2.5** にアップグレードされました。
- Spring Session は **2.2.2** にアップグレードされました。

### 6.3. RED HAT DATA GRID エンドポイントを開始および停止する機能

Red Hat Data Grid コマンドラインインターフェイス (CLI) は、Red Hat Data Grid エンドポイントコネクタを開始および停止するための以下のコマンドを提供します。

- **:stop-connector**
- **:start-connector**

詳細は、[Red Hat Data Grid エンドポイントの開始と停止](#) を参照してください。

### 6.4. HOT ROD アクセスログにタスク名が含まれる

リモート Hot Rod クライアントを介して Red Hat Data Grid サーバーでタスクを実行すると、Hot Rod アクセスログにタスク名が含まれるようになりました。

Hot Rod アクセスログの詳細については、[アクセスログ](#) を参照してください。

### 6.5. クロスサイトレプリケーションパフォーマンスの複数の向上

このリリースには、クロスサイトレプリケーションパフォーマンスの複数の向上が含まれています。現在の Red Hat Data Grid:

- 非同期レプリケーション要求を並行して処理します。
- スレッドプールを使用して、非同期レプリケーション要求を処理します。
- ブロックスレッドからサイト間でデータをバックアップする操作を送信します。

## 第7章 7.3.5 の更新と機能強化

### 7.1. クラスター化された最大アイドル有効期限

このリリースでは、クラスター化された **max-idle** 有効期限が拡張され、クラスター化されたキャッシュモードでのデータ損失が防止されます。

このリリース以降、クライアントが **max-idle** 有効期限値を持つエントリーを読み取ると、Red Hat Data Grid は `touch` コマンドをすべての所有者に送信します。これにより、エントリーの相対アクセス時間がクラスター全体で同じになります。

[最大アイドル有効期限](#) を参照してください。

### 7.2. ソケットタイムアウトの例外による接続の詳細の提供

このリリース以降、Hot Rod クライアント接続がタイムアウトすると、Red Hat Data Grid は、以下の例のように例外メッセージで接続に関する追加の詳細を提供します。

#### このリリースの前のソケットタイムアウト例外

```
org.infinispan.client.hotrod.exceptions.TransportException::  
java.net.SocketTimeoutException: ContainsKeyOperation{ExpirationCache,  
key=[B0x033E0131, flags=0] timed out after 60000 ms
```

#### このリリース以降のソケットタイムアウト例外

```
org.infinispan.client.hotrod.exceptions.TransportException::  
java.net.SocketTimeoutException: ContainsKeyOperation{ExpirationCache,  
key=[B0x033E0131, flags=0, connection=127.0.0.1/127.0.0.1:11222} timed out  
after 60000 ms
```

### 7.3. HOT ROD クライアント接続プールの設定

このリリース以降、Data Grid Hot Rod C++ クライアントでは、接続プールが使い果たされると、回復ポリシーを設定できます。

- **configurationBuilder.connectionPool().exhaustedAction(WAIT)** は、接続が返されて利用可能になるまで、無期限に待機します。
- **configurationBuilder.connectionPool().exhaustedAction(EXCEPTION)** は、**NoSuchElementException** を出力します。

### 7.4. DATA GRID のドキュメント

Data Grid サーバーのドキュメントが更新され、Prometheus を使用して、JMX 統計を収集する方法が説明されました。

[Prometheus への JMX Bean の公開](#) を参照してください。

Data Grid ユーザーガイドが更新され、クロスサイトレプリケーションの容量係数の設定が明確になりました。



---

キーオーナーシップの「容量係数」を参照してください。

## 第8章 7.3.4 の更新と機能強化

### 8.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.5 へのアップグレード

Data Grid サーバーは EAP 7.2.5 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。

### 8.2. SPRING BOOT バージョンのアップグレード

Data Grid のこのリリースは、次のように Spring 5 依存関係を使用する Spring Boot 2.2.0 にアップグレードされます。

- Spring 5 は **5.2.0** にアップグレードされました。
- Spring Session は **2.2.0** にアップグレードされました。
- Spring Security は **5.2.0** にアップグレードされました。

### 8.3. クラスター化されたキャッシュを持つプライマリ所有者からの期限切れ

Data Grid は、エントリーが有効期限基準を満たしているとプライマリ所有者が判断した場合のみ、クラスター化されたキャッシュからのエントリーを期限切れにします。この変更により、重複する有効期限コマンドが減るため、パフォーマンスが向上します。

### 8.4. クロスサイトレプリケーションの有効期限コマンド

エントリーが期限切れになると、Red Hat Data Grid はクラスター間で期限切れのエントリーを削除するコマンドを複製します。このリリース以降、Red Hat Data Grid は、このコマンドをサイト間で複製しなくなり、パフォーマンスが向上します。

### 8.5. RED HAT DATA GRID CLI セッションの有効期限

CLI セッションが 6 分間のアイドルタイムアウト後に期限切れになるという事実に関するドキュメントが追加されました。詳細については、[コマンドラインインターフェイス \(CLI\) セクション](#)を参照してください。

## 第9章 7.3.3 の更新と機能強化

### 9.1. DATA GRID サーバーの JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.4 へのアップグレード

Data Grid サーバーは EAP 7.2.4 にアップグレードされ、パフォーマンスが向上し、セキュリティを強化するための複数の CVE 修正が含まれています。

### 9.2. AMAZON WEB SERVICES (AWS) のネイティブ S3 PING

AWS S3 API への変更により、Data Grid は JGroups **S3\_PING** プロトコルの代わりに **NATIVE\_S3\_PING** を使用する必要があります。

JGroups **S3\_PING** は、AWS でのサーバー検出には機能しなくなりました。既存の **S3\_PING** 設定を **NATIVE\_S3\_PING** に移行する必要があります。

詳細は、以下を参照してください。

- Red Hat Data Grid ユーザーガイドの [Amazon Web Services](#)。
- Red Hat Data Grid サーバーガイドの [JGroups サブシステム設定](#)。
- [SigV2 の廃止に関する Amazon S3 の更新](#)

### 9.3. JGROUPS トランスポートスタックでのフロー制御の変更

このリリースには、ライブラリーモードでの Data Grid クラスターのトランスポート層のパフォーマンスを向上させる変更が含まれています。

- Data Grid は、JGroups Unicast Flow Control (**UFC**) をデフォルトの **TCP** トランスポートスタックに追加します。**UFC** は、受信ノードの JGroups スレッドプールが最大容量に達すると、ノードが他のノードに非同期メッセージを送信しないようにします。これが発生すると、受信ノードはメッセージを破棄し始めますが、送信ノードはメッセージを再送信し続けます。
- **UFC** および **MFC** の **max\_credits** 属性の値は、JGroups のデフォルトの **2m** から **3m** に増加します。この属性は、受信ノードからの確認応答なく、あるノードが別のノードに送信できる最大バイト数を設定します。



#### 注記

- **UFC** は、ライブラリーモードのデフォルトの **UDP** トランスポートスタックにすでに含まれています。
- サーバーモードでは、Data Grid は非ブロックフロー制御 (**UFC\_NB** および **MFC\_NB**) をデフォルトとして使用します。

### 9.4. REST エンドポイント承認

Data Grid REST エンドポイントは、承認用に設定されたキャッシュへのアクセスを許可するようになりました。

この機能強化により、この [KCS ソリューション](#) で概説されている問題が解決されます。

## 9.5. PROTOSTREAM ライブラリーの 4.2.4.FINAL への更新

Data Grid の Protostream ライブラリーコンポーネントが 4.2.4.Final に更新されました。

## 第10章 7.3.2 の更新と機能強化

### 10.1. 非同期クロスサイトレプリケーションでサイトを自動的にオフラインにする

Data Grid は、非同期レプリケーション戦略でクロスサイトレプリケーション機能を使用する場合、**take-offline** 設定を適用するようになりました。

### 10.2. JMX による非同期クロスサイトレプリケーション統計

Data Grid は、非同期クロスサイト複製について次の統計を追加します。

- **AsyncXSiteAcksCount**
- **AverageAsyncXSiteReplicationTime**
- **MaximumAsyncXSiteReplicationTime**
- **MinimumAsyncXSiteReplicationTime**

詳細については、[JMX コンポーネント](#) ドキュメントを参照してください。

### 10.3. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.1 との依存関係の調整

サーバーモードの Data Grid 7.3.2 のコアモジュールの依存関係は、EAP 7.2.1 に一致します。

### 10.4. リスナー通知のスキップ

Hot Rod クライアントに **SKIP\_LISTENER\_NOTIFICATION** フラグが含まれるようになったため、セッション ID が変更された際、クライアントリスナーが Data Grid サーバーから通知を受け取らなくなりました。

このフラグは、Spring 5 との **spring-session** 統合を使用する際の問題を解決します。Spring 5 で **spring-session** を使用している場合は、Data Grid サーバーと Hot Rod クライアントの両方を 7.3.2 にアップグレードする必要があります。

同様に、**SKIP\_LISTENER\_NOTIFICATION** フラグを設定する前に、Data Grid サーバーと Hot Rod クライアントの両方を 7.3.2 以降にアップグレードする必要があります。

詳細については、Data Grid ユーザーガイドの [通知のスキップ](#) を参照してください。

## 第11章 7.3.1 の更新と機能強化

### 11.1. JAVA 11

Data Grid が Java 11 をサポートするようになりました。サポートされる構成については、<https://access.redhat.com/articles/2435931> を参照してください。



#### 注記

Data Grid for OpenShift リリース 7.3.1 は、Java 8 で構築されています。Data Grid for OpenShift は、バージョン 7.3.2 で Java 11 をサポートします。詳細は、サポートされる構成のページを参照してください。

### 11.2. OPENJDK 8 FOR MICROSOFT WINDOWS

Data Grid は、Windows で OpenJDK 8 をサポートするようになりました。サポートされる構成については、<https://access.redhat.com/articles/2435931> を参照してください。

### 11.3. JGROUPS の 4.0.18.FINAL への更新

Data Grid は、JGroups バージョン 4.0.18.Final を使用するようになりました。コンポーネントの詳細については、<https://access.redhat.com/articles/488833> を参照してください。

### 11.4. JGROUPS DNS\_PING

Data Grid は、クラスター検出に JGroups **DNS\_PING** プロトコルを使用できるようになりました。



#### 注記

Data Grid for OpenShift イメージは、JGroups **DNS\_PING** と同じ機能を提供する **openshift.DNS\_PING** を使用します。デフォルトでは、Data Grid for OpenShift イメージで JGroups **DNS\_PING** を有効にすることはできません。ただし、JGroups **DNS\_PING** を使用するカスタム設定を使用して、Data Grid for OpenShift イメージからカスタムイメージを構築できます。カスタム OpenShift アプリケーションに Data Grid を組み込む場合は、JGroups **DNS\_PING** も使用できます。

詳細については、[JGroups サブシステムの設定](#) を参照してください。

### 11.5. DATA GRID FOR OPENSIFT の設定可能なログレベル

このリリースでは、Data Grid がログメッセージをキャプチャーするカテゴリとレベルを調整する **LOGGING\_CATEGORIES** 環境変数が追加されています。詳細については、[監視とログ](#) を参照してください。

### 11.6. ファイル名アナライザー

Data Grid は、デフォルトの **filename** アナライザーを提供するようになりました。詳細については、[デフォルトのアナライザー](#) を参照してください。

## 第12章 7.3.0 の新機能と機能強化

### 12.1. DATA GRID FOR OPENSIFT の向上

このリリースには、Data Grid for OpenShift の複数の向上が含まれています。

- キャッシュサービス (**cache-service**) の完全なサポート。
- Data Grid for OpenShift の完全なディストリビューションを提供する Data Grid サービス (**datagrid-service**)。
- Data Grid for OpenShift イメージの機能強化。
- Prometheus との統合による監視機能。
- OpenShift で実行されているコンテナ化されたアプリケーションに Data Grid を埋め込むことができるライブラリーモードのサポート。制限が適用されます。詳細については、[Data Grid for OpenShift](#) を参照してください。



#### 注記

Red Hat は、Data Grid をカスタムサーバーアプリケーションに埋め込むことを推奨していません。Data Grid でクライアントアプリケーションからのキャッシュ要求を処理する場合は、Data Grid for OpenShift をデプロイします。

詳細を確認し、開始するには、[Data Grid for OpenShift ドキュメンテーション](#) にアクセスしてください。

### 12.2. フレームワークの統合

Data Grid のこのリリースでは、Spring や Hibernate などの有名なエンタープライズ Java フレームワークとの統合が向上しています。

#### 12.2.1. Spring の機能強化

このリリースでは、Spring Framework との Data Grid 統合に複数の機能強化が追加されています。



#### 注記

Data Grid のこのリリースは、Spring Framework および Spring Boot の特定のバージョンをサポートします。サポートされる構成については、<https://access.redhat.com/articles/2435931> を参照してください。

##### 12.2.1.1. 取得操作の同期

Data Grid は [SPR-9254](#) を実装するため、**get()** メソッドが、マルチスレッドメカニズムを使用して、キー値を返します。

詳細については、**org.springframework.cache** インターフェイスの [get](#) の説明を参照してください。

##### 12.2.1.2. 非同期操作とタイムアウトの設定

このリリースでは、非同期操作とタイムアウトの設定に関する変更がいくつかあります。

Data Grid を Spring キャッシュプロバイダーとして使用する際、読み取り操作と書き込み操作を待機する最大時間を設定できるようになりました。タイムアウトにより、メソッド呼び出しを非同期で行うことができます。

次の **put()** メソッドの例を使用して、タイムアウトの前後の違いを検討してください。

### 書き込みタイムアウト前

```
public void put(Object key, Object value, long lifespan, TimeUnit unit) {
    this.cacheImplementation.put(key, value != null ? value : NullValue.NULL, lifespan, unit);
}
```

### 書き込みタイムアウト後

```
public void put(final Object key, final Object value) {
    this.nativeCache.put(key, value);
    try {
        if (writeTimeout > 0)
            this.nativeCache.putAsync(key, value != null ? value : NullValue.NULL).get(writeTimeout,
                TimeUnit.MILLISECONDS);
        else
            this.nativeCache.put(key, value != null ? value : NullValue.NULL);
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
        throw new CacheException(e);
    } catch (ExecutionException | TimeoutException e) {
        throw new CacheException(e);
    }
}
```

書き込み操作のタイムアウトを設定すると、**putAsync** が呼び出されます。これは、他の書き込みをブロックしないファイアアンドフォーゲット呼び出しです。

タイムアウトを設定しない場合は、同期 **put** が呼び出され、他の書き込みがブロックされます。

**infinispan.spring.operation.read.timeout** および **infinispan.spring.operation.write.timeout** でタイムアウト設定を行います。方法については、ドキュメントの [タイムアウトの設定](#) を参照してください。

#### 12.2.1.3. Spring アプリケーションの集中設定プロパティー

Data Grid をリモートクライアントサーバーモードで Spring キャッシュプロバイダーとして使用している場合は、クラスパスの **hotrod-client.properties** で設定プロパティーを設定できます。その後、アプリケーションは、その設定で **RemoteCacheManager** を作成できます。

使用可能な設定プロパティーに関する情報は、[org.infinispan.client.hotrod.configuration](#) パッケージの [説明](#) で入手できます。

#### 12.2.1.4. キャッシュ名を取得する機能

**RemoteCacheManager** クラスには、**["cache1", "cache2"]** などの文字列の JSON 配列としてキャッシュ名を返す **getCacheNames()** メソッドが含まれるようになりました。このメソッドは **org.springframework.cache.CacheManager** 実装に含まれているため、Data Grid を Spring キャッシュプロバイダーとして使用する際、定義されたキャッシュ名を検索できます。



[RemoteCacheManager](#) の Javadoc で詳細を確認してください。

### 12.2.1.5. Spring Boot スターター

Data Grid には、Spring Boot スターターが含まれており、迅速に起動および実行できます。[Data Grid Spring Boot スターター](#) を参照してください。

### 12.2.2. Hibernate のセカンドレベル (L2) キャッシング

Data Grid は、アプリケーションの永続レイヤーのパフォーマンスを向上させるために、(L2) キャッシュプロバイダーとして Hibernate とシームレスに統合されています。

Hibernate は、Java 用のオブジェクト/リレーショナルマッピング (ORM) 機能を提供し、完全に準拠した JPA (Java Persistence API) 永続化プロバイダーです。Hibernate は、キャッシュ内のオブジェクトがセッションにバインドされる第1レベル (L1) キャッシュを使用します。L2 キャッシュプロバイダーである Data Grid は、すべてのセッションでオブジェクトのグローバルキャッシュとして機能します。

次の場所で、Data Grid を L2 キャッシュとして設定できます。

- JPA: `persistence.xml`
- Spring: `application.properties`

L2 キャッシュの有効化に関する完全な情報、およびさまざまなデプロイメントシナリオの情報については、ドキュメントの [JPA/Hibernate L2 Cache](#) を参照してください。

### 12.2.3. 組み込みモードでの Red Hat SSO との統合

このリリースでは、Red Hat SSO を使用して、**ライブラリー (組み込み) モード** で Data Grid へのアクセスを保護するためのサポートを提供します。

詳細については、[secure-embedded-cache](#) クイックスタートを参照し、Red Hat SSO との統合を示すサンプルアプリケーションをデプロイおよび実行してください。

## 12.3. HOT ROD クライアントの向上

### 12.3.1. トランザクションのサポート

Java、C++、および C# の Hot Rod クライアントは、トランザクションを開始し、参加できるようになりました。

Java Hot Rod クライアントは、**FULL\_XA** と **NON\_XA** の両方のトランザクションモードをサポートしています。C++ および C# Hot Rod クライアントは、**NON\_XA** トランザクションモードのみをサポートします。

詳細については、[Hot Rod トランザクション](#) を参照してください。

### 12.3.2. JMX での Java Hot Rod クライアント統計

**ServerStatistics** インターフェイスは、JMX を介して Java Hot Rod クライアントの統計を公開するようになりました。

次の例のように、Hot Rod クライアント実装で JMX 統計を有効にする必要があります。

```

ConfigurationBuilder builder = new ConfigurationBuilder();
builder.addServer()
    .host("127.0.0.1")
    .port(11222)
    .statistics()
    .enable()
    .jmxEnable();

```

- **enabled()** を使用すると、クライアント側の統計を収集できます。
- **jmxEnabled()** は、JMX を介して統計を公開します。

詳細については、Javadoc の [ServerStatistics](#) を参照してください。

### 12.3.3. Java Hot Rod クライアント設定の機能強化

このリリースでは、**hotrod-client.properties** ファイルを使用して、Java Hot Rod クライアントの設定が向上しています。ニアキャッシュ設定、クロスサイト (xsite) プロパティ、認証と暗号化を制御する設定などを行うことができます。

詳細については、[Hot Rod クライアント設定の概要](#) を参照してください。

### 12.3.4. Netty に基づく新しい Java Hot Rod クライアントの実装

Hot Rod Java クライアントは Netty フレームワークを使用して構築されており、パフォーマンスが向上し、同じ接続を介して操作を同時に実行するためのサポートが提供されます。

以前のリリースでは、アプリケーションが実行する操作は同期的に行われるか、専用のスレッドプールに委譲されていました。

このリリース以降、操作は非同期で実行され、応答は **HotRod-client-async-pool** スレッドプールで処理されます。操作は同じ接続で多重化できるため、必要な接続数が少なくなります。



#### 注記

カスタムマーシャラーは、データの非整列化を呼び出す特定のスレッドに依存しないようにする必要があります。

詳細については、次の Javadoc を参照してください。

- [org.infinispan.client.hotrod.configuration.ConnectionPoolConfigurationBuilder](#)
- [org.infinispan.client.hotrod.configuration.Configuration](#)
- [org.infinispan.client.hotrod.configuration.ConfigurationBuilder](#)
- [ConfigurationChildBuilder](#)

### 12.3.5. JSON オブジェクトに対する Javascript Hot Rod クライアントのサポート

node.js Hot Rod クライアントは、キーおよび値としてネイティブ **JSON** オブジェクトのサポートを追加します。以前のリリースでは、クライアントは **String** キーおよび値のみをサポートしていました。

ネイティブ **JSON** オブジェクトを使用するには、クライアントを次のように設定する必要があります。

-

```

var infinispanspan = require('infinispanspan');
var connected = infinispanspan.client(
  {port: 11222, host: '127.0.0.1'}
  , {
    dataFormat : {
      keyType: 'application/json',
      valueType: 'application/json'
    }
  }
);
connected.then(function (client) {
  var clientPut = client.put({k: 'key'}, {v: 'value'});
  var clientGet = clientPut.then(
    function() { return client.get({k: 'key'}); });
  var showGet = clientGet.then(
    function(value) { console.log("get({k: 'key'})=" + JSON.stringify(value)); });
  return showGet.finally(
    function() { return client.disconnect(); });
}).catch(function(error) {
  console.log("Got error: " + error.message);
});

```

キーと値のデータ型を個別に設定できます。たとえば、キーを **String** として設定し、値を **JSON** として設定できます。



#### 注記

スクリプトは現在、ネイティブ **JSON** オブジェクトをサポートしていません。

### 12.3.6. Hot Rod によるキャッシュ名の取得

このリリースには、宣言またはプログラムによって定義されたキャッシュ名のコレクション、および **RemoteCacheManager** を介して実行時に作成されたキャッシュを返す **getCacheNames()** メソッドが含まれています。

Hot Rod プロトコルには、["cache1", "cache2"] などの文字列の JSON 配列としてキャッシュ名を返す **@@cache@names** 管理タスクも含まれるようになりました。

詳細は以下を参照してください。

- [Hot Rod 管理タスク](#)
- [RemoteCacheManager](#)

## 12.4. 永続性の向上

### 12.4.1. ライトビハインドキャッシュストアのフォールトトレランス

Data Grid では、ライトビハインドキャッシュストアを設定できるようになりました。これにより、ライトビハインド操作が失敗した場合、キャッシュでの追加の操作は許可されません。さらに、キャッシュストアへの書き込みに失敗した変更は、基になるキャッシュストアが使用可能になるまで、キューに入れられます。

**connection-attempts**、**connection-interval**、および **fail-silently** を宣言的に使用して、フォールトトレランスを設定できます。

プログラムでフォールトトレランスを設定するには、次を使用します。

- **PersistenceConfiguration** クラスの **connectionAttempts()** および **connectionInterval()** メソッド。
- **AsyncStoreConfiguration** クラスの **failSilently()** メソッド。

詳細については、次のドキュメントを参照してください。

- [ライトスルーおよびライトビハインドキャッシュ](#)
- [org.infinispan.configuration.cache.AsyncStoreConfiguration](#)
- [org.infinispan.configuration.cache.PersistenceConfiguration](#)
- [org.infinispan.notifications.cachelistener.annotation.PersistenceAvailabilityChanged](#)
- [設定スキーマ](#)

## 12.5. データの相互運用性

Data Grid は、さまざまなエンドポイントに適したフォーマット間でデータを変換するトランスコーディング機能を提供します。たとえば、Hot Rod エンドポイントを介して ProtoBuf でエンコードされたデータを書き込み、REST エンドポイントを介して JSON ドキュメントとして取得できます。



### 注記

互換モードは廃止され、Data Grid から削除されます。代わりに、プロトコルの相互運用機能を使用する必要があります。

## 12.6. デフォルトのアナライザー

Data Grid には、入力データを索引付けおよびクエリーできる1つ以上の用語に変換するデフォルトのアナライザーのセットが含まれています。

詳細については、[分析](#) を参照してください。

## 12.7. メトリックの向上

このリリースでは、JMX を介して新しい操作とメトリックが公開されます。

### ClusteredLockManager コンポーネント

- **forceRelease** は、ロックを強制的に解放します。
- **isDefined** は、ロックが定義されている場合、**true** を返します。
- ロックが存在し、取得されている場合、**isLocked** は、**true** を返します。
- **remove** はクラスターからロックを削除します。再びアクセスするには、ロックを再作成する必要があります。

## Passivation コンポーネント

- **passivateAll** は、すべてのエントリーを **CacheStore** にパッシベートします。

## CacheStore コンポーネント

- **NumberOfPersistedEntries** は、期限切れのエントリーを除く、現在保持されているエントリーの数を返します。

詳細については、[jmxComponents](#) を参照してください。

## 12.8. ストリームのロック

**LockedStream** インターフェイスの **invokeAll()** メソッドは、それぞれのキーに対してロックが保持されているエントリーのコードを実行する方法を提供するようになりました。これにより、ロックを取得する必要がなくなり、値の状態を保証できるようになります。

詳細については、[org.infinispan.LockedStream](#) を参照してください。

## 12.9. 設定の向上

### 12.9.1. 設定のワイルドカード

Data Grid が一致するキャッシュにテンプレートを適用できるように、設定テンプレート名にワイルドカードを使用できるようになりました。

詳細については、[キャッシュ設定のワイルドカード](#) を参照してください。

### 12.9.2. 不変の設定

不変の設定ストレージプロバイダーを作成して、キャッシュの作成または削除を防止できるようになりました。

グローバル状態で **IMMUTABLE** 設定ストアを宣言するか、プログラムで使用して、Data Grid を設定する場合は、**immutable-configuration-storage** パラメーターを使用します。

- [キャッシュの取得](#)
- [org.infinispan.configuration.global.GlobalStateConfiguration](#)
- [設定スキーマ](#)

## 12.10. キャッシュ操作のセキュリティーの向上

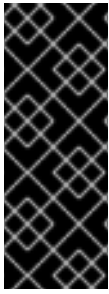
**AdvancedCache** インターフェイスには、キャッシュで許可が有効になっている場合、指定されたサブジェクトで操作を実行する **withSubject()** メソッドが含まれるようになりました。

**AdvancedCache** インターフェイスについては、Javadoc を参照してください。

## 12.11. HTTP/2 のサポート

Data Grid は、REST エンドポイントで HTTP/2 をサポートするようになりました。

## 第13章 DATA GRID 7.3 のテクノロジープレビュー



### 重要

テクノロジープレビュー機能は、Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされておらず、機能的に完全でない可能性があります。Red Hat は、実稼働環境での使用は推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

### 13.1. RED HAT OPENSIFT でのクロスサイトレプリケーション

Red Hat Data Grid for OpenShift は、異なるデータセンターを実行しているクラスター間でデータをバックアップするクロスサイトレプリケーション機能を提供します。

#### サポートされるアーキテクチャーと機能

現在、Red Hat OpenShift でのクロスサイトレプリケーションは、次のアーキテクチャーでサポートされています。

- datagrid-service サービステンプレートを使用して、**StatefulSet** としてデプロイされた Data Grid。
- 1つのマスターサイト。
- 1つのバックアップサイト。
- Data Grid ノードが JGroups **RELAY2** プロトコルを介して通信し、クロスサイトレプリケーションを実行するためのポートを公開する **NodePort** サービス。
- Data Grid は、異なるサイトでの同時変更に対する制御を提供しません。必要に応じて同時実行を処理するようにアプリケーションを設定するか、回避戦略を実装する必要があります。
- 監視および管理用の Data GridJMX コンポーネント。[JMX コンポーネント](#) を参照してください。

#### 現在サポートされていない特徴と機能

Data Grid は現在、Red Hat OpenShift のクロスサイトレプリケーションで次の特徴と機能をサポートしていません。

- スキーマの複製。
- データの索引付け。
- 動的キャッシュの作成。

#### リソースとドキュメント

[クロスサイトレプリケーション用の standalone.xml](#) は、Red Hat Data Grid for OpenShift の推奨設定を提供します。

GitHub 上の [Cross-Site Replication: Red Hat Data Grid for OpenShift](#) では、ドキュメントとクイックスタートチュートリアルが提供されています。

クロスサイトレプリケーションドキュメントには、あるサイトから別のサイトに状態を転送する手順など、追加の詳細が記載されています。

## 13.2. C++ および C# クライアントによるリモート管理

C++ および C# Hot Rod クライアントは、管理操作をリモートで実行できる **RemoteCacheManagerAdmin** インターフェイスの実装を提供するようになりました。

## 13.3. C++ および C# クライアントを使用したクラスター化されたカウンター

C++ および C# の Hot Rod クライアントは、クラスター化されたカウンターをリモートで操作する機能を提供するようになりました。

## 13.4. 管理コンソールの機能

Data Grid 管理コンソールでは、エンドポイントを設定および管理できるようになり、データを操作するための次の機能が提供されます。

- データのクエリー
- キャッシュエントリーの作成と更新
- キャッシュエントリーの削除

さらに、管理コンソールには、基本的な protobuf スキーマエディターが含まれています。

## 第14章 DATA GRID 7.3 で廃止された特徴と機能

Data Grid のこのリリースでは、さまざまな特徴と機能コンポーネントが廃止されています。

非推奨の機能は、Data Grid 7.x のサポートが終了するまで、引き続きサポートされます。非推奨の機能は、今後のメジャーリリースではサポートされず、新しいデプロイメントには推奨されません。



### 注記

この情報は執筆時点のもので、[JDG-1978](#) を使用して、次のメジャーリリースの非推奨コードをすべて確認できます。

### Hot Rod v1.x

Hot Rod プロトコルバージョン 1.x は、次のメジャーリリースではサポートされません。

### RocksDB が LevelDB キャッシュストアを置き換える

このリリースの時点で、LevelDB キャッシュストアは廃止され、RocksDB キャッシュストアに置き換えられました。LevelDB キャッシュストアにデータが保存されている場合、RocksDB キャッシュストアは、最初の実行時、そのデータを SST ベースの形式に変換します。

RocksDB は、特に高度な同時実行シナリオで優れたパフォーマンスと信頼性を提供します。ドキュメントで [RocksDB キャッシュストア](#) の詳細を確認してください。

### 互換性モード

互換モードは推奨されておらず、次のメジャーリリースで削除される予定です。複数のエンドポイントからキャッシュにアクセスするには、データをバイナリー形式で保存し、キーと値の **MediaType** を設定する必要があります。詳細については、以下のトピックを参照してください。

- [プロトコルの相互運用性](#)
- [MediaType](#)  
データをマーシャリングされていないオブジェクトとして保存する場合は、オブジェクトのコンテンツを保存するように、キーと値を次のように設定する必要があります。

```
<encoding>
  <key media-type="application/x-java-object"/>
  <value media-type="application/x-java-object"/>
</encoding>
```

### クラスター化されたエグゼキューターが分散エグゼキューター API を置き換える

Data Grid は、分散エグゼキューターをクラスター化されたエグゼキューター (クラスター内で任意のコードを実行するためのユーティリティー) に置き換えます。参照:

- [クラスターエグゼキューター](#)
- [org.infinispan.manager.ClusterExecutor](#)

### RemoteCache getBulk()

**getBulk()** メソッドは、**RemoteCache** インターフェイスで非推奨になりました。参照: [org.infinispan.client.hotrod.RemoteCache](#)。

### Agroal PooledConnectionFactory が c3p0/HikariCP JDBC PooledConnectionFactory を置き換える

JDBC **PooledConnectionFactory** は、**c3p0.properties** および **hikari.properties** で設定する接続プールを提供します。次のメジャーリリースでは、Data Grid は、Agroal と互換性のあるプロパ



ティーファイルのみを使用して設定する **PooledConnectionFactory** を提供します。 [Agroal プロジェクト](#) を参照してください。

#### CLI ローダー

**infinispan-persistence-cli** は非推奨になり、次のメジャーバージョンで削除される予定です。

#### 非推奨のクラス

- **org.infinispan.lifecycle.AbstractModuleLifecycle**
- **org.infinispan.lifecycle.Lifecycle**

#### Eager ニアキャッシュの残りのコード

Eager ニアキャッシュ機能の残りのコードアーティファクトは、次のメジャーバージョンで削除されます。

## パート II. サポートされる構成とコンポーネントのバージョン

## 第15章 DATA GRID 7.3 でサポートされる構成

Red Hat は、<https://access.redhat.com/articles/2435931> で入手可能な Red Hat Data Grid 7.3 の特定のハードウェアとソフトウェアの組み合わせをサポートします。

## 第16章 DATA GRID コンポーネントのバージョン

Red Hat Data Grid 7.3 のコンポーネントの詳細は、[カスタマーポータル](#) で入手できます。

次のように、プロジェクトから Data Grid コンポーネントのバージョンを見つけることもできます。

1. Data Grid ソースを含むアーカイブ **jboss-datagrid-`{version}`-sources.zip** を [Product Downloads](#) ページからダウンロードします。
2. アーカイブをローカルのファイルシステムにデプロイメントします。
3. ターミナルウィンドウを開き、**pom.xml** ファイルが含まれる最上位のディレクトリーに移動します。
4. **mvn dependency:tree** を実行し、依存関係に関する情報を取得します。

## パート III. 既知の問題および修正された問題

## 第17章 DATA GRID 7.3 の既知の問題

### Kerberos 認証を使用する場合の Java 8 での Hot Rod Java クライアントセキュリティ例外

**問題:** [JDG-4224](#)

**説明:** GSSAPI セキュリティ機能を備えた Java 8 で Hot Rod Java クライアントを使用すると、次の例外が出力されます。

```
org.infinispan.client.hotrod.exceptions.HotRodClientException: javax.security.sasl.SaslException: ELY05123: No security layer selected but message length received
```

この問題は、EAP 7.3 の Elytron サブシステムを使用した Kerberos 認証のより厳しい要件が原因で発生します。

**回避策:** Data Grid サーバー設定の **sasl** セキュリティーレルムに **wildfly.sasl.relax-compliance** プロパティを追加します。

```
<authentication security-realm="ApplicationRealm">
  <sasl server-context-name="hotrod-service" server-name="node0" mechanisms="GSSAPI"
  qop="auth" strength="high medium low">
    <policy>
      <no-anonymous value="true" />
    </policy>
    <property name="wildfly.sasl.relax-compliance">true</property>
  </sasl>
</authentication>
```

### 内部シリアライズライブラリーにより、JSON から Java オブジェクトへの変換が防止される

**問題:** [JDG-3965](#)

**説明:** アンマーシャリングされた Plain Old Java Objects (POJO) としてデータを Data Grid サーバーに保存し、JSON 形式でデータを読み書きしようとする、次の例外が発生します。

```
com.fasterxml.jackson.databind.exc.InvalidDefinitionException: Illegal type
(com.example.MyClass) to deserialize: prevented for security reasons
```

**回避策:** Data Grid サーバーで、次のシステムプロパティを使用して、ユーザークラスを指定します。

```
-Djackson.deserialization.whitelist.packages=com.example
```

次のように、完全修飾クラス名を指定します。

```
-Djackson.deserialization.whitelist.packages=com.example.MyClass
```

### OpenShift ConfigMap クイックスタートの一部のバージョンでの古いログ設定

**問題:** [JDG-4024](#)

**説明:** Data Grid **ConfigMap** クイックスタート [Customizing Data Grid Service Deployments](#) は、バージョン 7.3.4、7.3.5、および 7.3.6 では期待どおりに機能しません。

Data Grid for OpenShift 7.3.4 以降は、更新されたログフォーマッターを使用しますが、これは **ConfigMap** クイックスタートのログフォーマッターと互換性がありません。その結果、カスタマイズされた Data Grid サーバーイメージをクイックスタートで開始することはできません。

**回避策:** クイックスタートに 7.3.7 タグを使用するか、**7.3.x** ブランチで作業してください。

MySQL および PostgreSQL キャッシュストアドライバーは、IBM Z または IBM Power 上の OpenShift Container Platform の Data Grid for OpenShift イメージでは使用できません

**問題:** [JDG-3376](#)

**説明:** IBM Z または IBM Power 上の OpenShift Container Platform の Data Grid for OpenShift イメージは、MySQL および PostgreSQL キャッシュストアのドライバーを提供しません。

**回避策:** この問題の回避策はありません。

IBM Z または IBM Power 上の OpenShift Container Platform の Data Grid for OpenShift イメージを使用したプローブからの接続エラーイベント

**問題:** [JDG-3395](#)

**説明:** IBM Z または IBM Power 上の OpenShift Container Platform の Data Grid for OpenShift で、**Liveness probe failed:** および **Readiness probe failed:** というエラーメッセージが記録されません。

**回避策:** エラーメッセージを無視してください。イメージの準備ができると、停止します。

Data Grid コマンドラインインターフェイスで `ClassCastException` が発生する

**問題:** [JDG-3348](#)

**説明:** クロスサイトレプリケーション用に設定された Data Grid クラスター内のコントローラーノードを停止し、Data Grid CLI で **site** コマンドを実行すると、次の例外が発生します。

```
org.infinispan.remoting.responses.CacheNotFoundResponse cannot be cast to  
org.infinispan.remoting.responses.SuccessfulResponse
```

**回避策:** **site** コマンドを実行する前に、Data Grid ノードがクラスタービューから完全に削除されるまで待ちます。

テストの失敗により、Red Hat Fuse 6 を使用した Data Grid 7.3 の認定は不可能

**問題:** [JDG-2758](#)

**説明:** テストが失敗するため、Red Hat Fuse 6 を使用した Data Grid 7.3 の認定は不可能です。

**回避策:** この問題の回避策はありません。

Data Grid 7.3 を Red Hat Fuse 6 および 7 で Java 11 と組み合わせて使用できない

**問題:** [JDG-2800](#)

**説明:** Red Hat Fuse 6 および 7 は Java 11 と互換性がありません。その結果、Data Grid 7.3 を Red Hat Fuse 6 および 7 で Java 11 と組み合わせて使用することはできません。

**回避策:** この問題の回避策はありません。

EAP 6 および MySQL 5 を使用する Data Grid 7.3 で予期しない動作が発生する可能性がある

**問題:** [JDG-2871](#)

**説明:** EAP 6 および MySQL 5 で Data Grid を実行すると、予期しない動作が発生する可能性があります。

**回避策:** この問題の回避策はありません。

### Data Grid 7.3 ライブラリーが EAP 6 または Oracle WebLogic に正常にデプロイされない

問題: [JDG-2559](#)

説明: Data Grid 7.3 ライブラリーは、EAP 6 または Oracle WebLogic に正常にデプロイされません。

回避策: この問題の回避策はありません。

### Data Grid 6.6.2 からのローリングアップグレードが失敗する

問題: [JDG-2832](#)

説明: Data Grid 6.6.2 から Data Grid 7.3 へのローリングアップグレードを実行しようとする、例外が発生し、データが正常に移行されません。

回避策: この問題の回避策はありません。

### JBoss Web Server から Data Grid への HTTP セッションの外部化が FINE 永続化戦略で機能しない

問題: [JDG-2796](#)

説明: `persistenceStrategy` 属性を `FINE` の値に設定すると、HTTP セッションの外部化が予期しない動作をする原因になります。

回避策: `persistenceStrategy` 属性を `COARSE` の値に設定します。

### RocksDB キャッシュストアが Red Hat Enterprise Linux 6 または Microsoft Windows プラットフォームでサポートされない

問題: [JDG-2761](#)

説明: 現在、Red Hat Enterprise Linux 6 または Microsoft Windows プラットフォームで RocksDB キャッシュストアを使用することはできません。

回避策: この問題の回避策はありません。

### GLIBC\_2.14 Red Hat Enterprise Linux (RHEL) 6 の RocksDB キャッシュストアでのエラー

問題: [JDG-2546](#)

説明: RHEL 6 で RocksDB キャッシュストアを作成する際、エラー `/lib64/libc.so.6: version `GLIBC_2.14' not found` が発生しました。

回避策: この問題の回避策はありません。

### 認証が有効な場合、SKIP\_CACHE\_LOAD フラグは効果がない

問題: [JDG-1424](#)

説明: リモートクライアント/サーバーモードでは、キャッシュストア設定で `SKIP_CACHE_LOAD` フラグを設定し、Hot Rod クライアントで認証を有効にすると、エビクトされたエントリーを含むすべてのエントリーがキャッシュから取得されます。

回避策: この問題の回避策はありません。

### リロードが必要な状態の Data Grid 管理コンソールで無効になっているクラスターアクション

問題: [JDG-1843](#)

説明: 設定の変更後、クラスターの再起動を選択した場合、Data Grid クラスターに使用可能なアクションが管理コンソールで使用できません。この場合、クラスターは `Reload-Required` 状態にあります。



**Reload** および **Stop** アクションは、クラスター内の各ノードで使用できます。

**回避策:** クラスター内の1つ以上のノードをリロードして、クラスターレベルでアクションを復元します。

#### Data Grid 管理コンソールからエビクションストラテジーを変更すると、エラーが発生する

**問題:** [JDG-1804](#)

**説明:** Data Grid がドメインモードで実行されており、管理コンソールを使用して設定のエビクションストラテジーを変更した場合、変更を適用するために再起動しないと、エラーが発生します。

**回避策:** エビクションストラテジーを変更した後、サーバーを再起動します。

#### クラスター間のローリングアップグレード中、断続的なデータ損失が発生する

**問題:** [JDG-991](#)

**説明:** Data Grid のローリングアップグレードを実行する場合、ソースクラスター内のノードがオフラインになった後、移行されたすべてのデータがターゲットクラスターから削除される可能性があります。

**回避策:** この問題の回避策はありません。

#### Data Grid 7.0 以前で永続ストレージからデータを読み取る際、NullPointerException が発生する

**問題:** [JDG-968](#)

**説明:** Data Grid 7.0 以前でキャッシュストアにデータを保存し、Data Grid 7.1 以降でそのデータを読み取ろうとすると、エラーが発生し、データを読み取ることができません。



#### 注記

この問題は、Data Grid 7.1 から、以降のバージョンにアップグレードする場合は、当てはまりません。

**回避策:** この問題の回避策はありません。

## 第18章 修正された問題

### 18.1. DATA GRID 7.3.9 で修正済み

Red Hat Data Grid 7.3.9 には、以下の重要な修正が含まれています。

- [JDG-4160](#) HotRodSourceMigrator でのメモリーリーク
- [JDG-4550](#) セキュリティーマネージャーが有効な場合は、1つのファイルストアを使用するキャッシュが起動に失敗する
- [JDG-4857](#) 初期サーバーリストスイッチはトポロジーエージをインクリメントする必要がある
- [JDG-4956](#) 値のサイズが大きいエントリーを追加すると、Hot Rod クライアントでメモリーリークが発生する
- [JDG-4387](#) 統計が有効な単純なキャッシュによって、EvictionManagerImpl で NullPointerException が発生する
- [JDG-4767](#) Hot Rod クライアントのクラスター切り替えが早すぎる

### 18.2. DATA GRID 7.3.8 で修正済み

Red Hat Data Grid 7.3.8 には、以下の重要な修正が含まれています。

- [JDG-4151](#) EntryMergePolicyFactoryRegistry クラスを使用して、カスタム EntryMergePolicyFactory 実装を登録すると、ConcurrentModificationException が発生する

### 18.3. DATA GRID 7.3.7 で修正済み

Red Hat Data Grid 7.3.7 には、以下の重要な修正が含まれています。

- [JDG-3700](#) EAP 7.3 からセッションを外部化するアプリケーションでの予期しない動作。
- [JDG-3848](#) ノードがクラスターを離れると、SingleFileStore キャッシュストアで問題が発生する可能性がある。
- [JDG-3818](#) 登録済みリスナーが期待どおりに動作せず、メモリー不足エラーが発生する。
- [JDG-3498](#) 演算操作でリモートタスクを実行すると、エビクションが使用されている場合、エラーメッセージが表示される。
- [JDG-3497](#) 演算操作でリモートタスクを実行すると、オフヒープストレージが使用されている場合、エラーメッセージが表示される。

### 18.4. DATA GRID 7.3.6 で修正済み

Red Hat Data Grid 7.3.6 には、以下の重要な修正が含まれています。

- [JDG-2644](#) Data Grid 管理コンソールは、クラスター内の最初のノードを表示する。
- [JDG-3527](#) バイナリストレージを使用している場合は、クライアントリクエストが誤った Data Grid サーバーにルーティングされる可能性がある。

- [JDG-3522](#) JDBC 文字列ベースのキャッシュストアからデータをパージすると、SQL サーバーの例外が発生する。
- [JDG-3450](#) デプロイされたサーバースタックでアノテーション生成マーシャラーを使用すると、クラスロードの問題が発生する。
- [JDG-3529](#) クラスタートランスポートの JGroups サブシステムは、RELAY プロトコルのプロパティを保持しない。

## 18.5. DATA GRID 7.3.5 で修正済み

Red Hat Data Grid 7.3.5 には、以下の重要な修正が含まれています。

- [JDG-3355](#) RpcManager:SitesView 属性が空であるか、XSiteRepl メンバーの不完全なビューが含まれている。
- [JDG-3366](#) トランザクションキャッシュを使用すると、有効期限操作のパフォーマンスが低下する。
- [JDG-3354](#) クラスタ化されたキャッシュモードでノードフェイルオーバーが発生すると、最大アイドル設定により、データが誤って期限切れになる。
- [JDG-3413](#) 無効化コマンドは、キャッシュストアから以前の値を不必要にロードする。
- [JDG-3428](#) JDBC 文字列ベースのキャッシュストアの例が誤っている。
- [JDG-3357](#) HotRod クライアントは、ソケットタイムアウトを使用する操作のタイムアウトハンドラーを登録する。
- [JDG-3309](#) 非同期モードを使用する JPA キャッシュストアで設定されたキャッシュ。キャッシュエントリを変更すると、DEBUG 例外が発生し、キャッシュストアからエントリを削除する操作が成功しない。
- [JDG-2532](#) clustered.xml ファイルを使用して、サーバーを起動すると、管理コンソールがロードされない。
- [JDG-3416](#) ノードがクラスターに参加すると、保存されたインデックスが複製されない。

## 18.6. DATA GRID 7.3.4 で修正済み

Red Hat Data Grid 7.3.4 には、以下の重要な修正が含まれています。

- [JDG-3200](#) エントリが期限切れになると、Data Grid はサイト全体で期限切れのエントリを削除するコマンドを複製する。
- [JDG-3149](#) Prometheus モニタリングを有効にして、Data Grid for OpenShift をデプロイすると、Pod の起動に失敗しました。WFLYCTL0079 および WFLYLOG0078 例外がログに書き込まれました。
- [JDG-3264](#) FD\_ALL が非メンバーにメッセージを送信し、ログに警告メッセージが生成された。
- [JDG-3167](#) クラスタ化されたキャッシュ統計の JMX MBean が、例外のエビクションタイプで動作しなくなる。

- [JDG-3194](#) ISPN004034 メッセージが、ニアキャッシュ設定のクエリー時、ログに書き込まれる。
- [JDG-3214](#) 管理コンソールが応答しなくなり、キャッシュコンテナの許可を設定した場合、Data Grid ログファイルで例外が発生した。
- [JDG-3324](#) \*クロスサイト複製で発生したキャッシュ null 例外の誤ったサイトステータス
- [JDG-3185](#) 永続化設定でプリロードが有効になっている場合は、キャッシュエントリーの作成日が Data Grid サーバーの開始日に設定される

## 18.7. DATA GRID 7.3.3 で修正済み

Red Hat Data Grid 7.3.3 には、以下の重要な修正が含まれています。

- [JDG-3137](#) jboss-cli.xml ファイルで解析エラーが発生する。
- [JDG-3114](#) RocksDB ReadOptions メモリーリーク
- [JDG-2960](#) ターゲットノードがクラスター内にある場合は、CLI コマンドを使用した proto ファイルの追加が失敗する。
- [JDG-2834](#) キャッシュストアで複数のキャッシュを開始すると、カスタムキャッシュストアのスレッドの問題が発生する。
- [JDG-2148](#) クライアントが @Indexed アノテーションの不要な警告メッセージをログに記録する。
- [JDG-2117](#) 操作中、トランザクションキャッシュが停止すると、誤った値が生じる。
- [JDG-3106](#) Operator がキースタアを含むボリュームのマウントに失敗する。
- [JDG-2968](#) ReflectionUtil クラスに必要なセキュリティー更新

## 18.8. DATA GRID 7.3.2 で修正済み

Red Hat Data Grid 7.3.2 には、以下の重要な修正が含まれています。

- [JDG-2673](#) 共有ストアを持つトランザクション無効化キャッシュで古い読み取りが発生する。
- [JDG-2922](#) TCP: 完全な TCP 送信ウィンドウで send() がブロックされると、接続のクローズがブロックされる可能性がある。
- [JDG-2854](#) 複数のサーバースタア設定を区別することはできない。
- [JDG-2836](#) カウンターマネージャーの設定がサーバーに適用されていない。
- [JDG-2835](#) カウンタークライアントは、カウンターの検索に CH を使用しない。
- [JDG-2817](#) computelfAbsent での Stackoverflow エラー
- [JDG-2679](#) キャッシュがローカルの場合、共有ストアは例外を出力する必要がある。
- [JDG-2897](#) openshift.KUBE\_PING が OCP4.1 で機能しない。

- [JDG-2909](#) セッションのバグ。セッションは、セッション ID が変更された場合、削除される。

## 18.9. DATA GRID 7.3.1 で修正済み

Red Hat Data Grid 7.3.1 には、以下の重要な修正が含まれています。

- [JDG-2561](#) リモートクライアントサーバー Spring Boot サンプルアプリケーションでメトリックを使用できない。
- [JDG-2528](#) カスタムクラスパスでクイックスタート例外が発生する。
- [JDG-2534](#) WildFly モジュールのロードの失敗。JDK 8 で Data Grid を実行すると、モジュールのロードの失敗が発生する。
- [JDG-272](#) コマンドラインインターフェイススクリプトの名前は `ispn-cli.sh` である。
- [JDG-2529](#) キャッシュの作成後、管理コンソールが応答しなくなる。
- [JDG-2528](#) クイックスタートプロジェクトがカスタムクラスパスで失敗する。
- [JDG-2504](#) IBM JDK で実行すると、例外が発生する。
- [JDG-2518](#) ハッシュスペースセグメントが不十分なため、キャッシュインスタンスを開始できない。

## 18.10. DATA GRID 7.3.0 で修正済み

### Red Hat Data Grid 7.3.0 GA で解決されたすべての問題のリスト

機能および機能強化、ドキュメントの改善、バグ修正。

## パート IV. DATA GRID 7.3 への移行

Data Grid 7.3 への移行は、製品の変更点を確認し、既存の設定と使用方法を変更して、アップグレードに成功できます。

## 第19章 DATA GRID 7.3 の変更点

### 19.1. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP)

Data Grid サーバーは、このリリースの EAP 7.2 に基づいています。適切なバージョンの EAP ドキュメントを参照し、基礎となる EAP 設定またはセットアップが 7.2 でサポートされていることを確認してください。



#### 注記

Data Grid のこのリリースは、EAP 7.2 のみで **org.infinispan.extension** モジュールをサポートします。

このリリースでサポートされている EAP のバージョンについては、<https://access.redhat.com/articles/2435931> のされる構成を参照してください。

### 19.2. EAP モジュール

Data Grid は、EAP 用のライブラリーと Java Hot Rod クライアントモジュールを 1 つのパッケージとして提供するようになりました。以前のリリースでは、Data Grid はライブラリーモジュールと Java Hot Rod クライアントモジュールを個別のパッケージとして提供していました。[カスタマーポータル](#) から EAP のモジュールをダウンロードします。



#### 注記

Data Grid EAP モジュールは、**system/add-ons/ispn** ディレクトリーに置かれるようになりました。

### 19.3. JBOSS FUSE の CAMEL コンポーネント

Apache Camel 統合のために、Red Hat Fuse チームは **camel-jbosmdatagrid** コンポーネントに取って代わる **camel-infinispan** コンポーネントを積極的に開発および管理しています。Red Hat は、Red Hat Fuse 7.3 以降で利用可能な **camel-infinispan** コンポーネントを使用することを推奨します。

詳細については、[Red Hat Fuse のドキュメント](#) を参照してください。

### 19.4. キャッシュストアの互換性

Data Grid 7.3 では、以前のバージョンの Data Grid と後方互換性がない内部マーシャリング機能の変更が導入されています。その結果、Data Grid 7.3.x 以降では、以前のバージョンの Data Grid で作成されたキャッシュストアを読み取ることができません。さらに、Data Grid は、JDBC 混合およびバイナリストアなどの一部のストア実装を提供しなくなりました。

**StoreMigrator.java** を使用して、キャッシュストアを移行します。この移行ツールは、以前のバージョンのキャッシュストアからデータを読み取り、現在のマーシャリングの実装との互換性のために、コンテンツを書き換えます。

詳細については、[Store Migrator](#) を参照してください。

### 19.5. MEMCACHED ストレージ

Data Grid Memcached エンドポイントは、キーを **java.lang.String** として保存しなくなりました。互換性を高めるために、Data Grid はキーを UTF-8 エンコードされた文字列を表す **byte** 配列として保存します。

Memcached エンドポイントを使用する場合は、データをキャッシュにリロードして、キーを **byte** として保存する必要があります。ローリングアップグレードを実行するか、外部ソースからデータをロードします。

## 19.6. MEMCACHED コネクター

デフォルトの **memcache-connector** は、セキュリティーを考慮して、無効になっています。**memcache-connector** を有効にするには、エンドポイントサブシステムを設定する必要があります。[Memcached コネクターの設定](#) を参照してください。

## 19.7. スクリプトの応答

テキストベースのデータ型の分散スクリプトは、各サーバーからの結果が null の場合、**null** を返さなくなりました。応答は、個々の結果を含む JSON 配列になりました (例: **[null, null]**)。

## 19.8. サーバースレッドプール

子の Netty イベントループを処理するスレッドの名前が \*-ServerWorker から \*-ServerIO に変更されました。

## 19.9. パッシベーションを使用した共有キャッシュストア

キャッシュストアを共有として設定してから、ストアのパッシベーションを有効にすることはできません。この設定では、ノード間でデータを同期できないため、データの不整合が発生する可能性があります。このため、Data Grid は、この設定を検出すると、起動時に例外を出力するようになりました。

## 19.10. AFFINITYINDEXMANAGER を使用したデフォルトのシャード

デフォルトのシャード数は **4** になりました。以前のリリースでは、デフォルトのシャード数はキャッシュ内のセグメント数と同じでした。

## 19.11. ADVANCEDCACHELOADER の変更

**AdvancedCacheLoader** SPI により、Reactive Streams ベースの **publishKeys** および **publishEntries** メソッドが可能になり、パフォーマンス、スレッド化、および使いやすさが向上します。この変更は、カスタム **CacheLoader** 実装に影響します。

## 19.12. DATA GRID 7.3 での非推奨事項

このリリースでは、Data Grid 7.3 への移行に影響する特徴と機能が廃止されています。特に、キーと値のペアの **MediaType** を設定し、データをバイナリー形式で保存することにより、互換モードは廃止されます。LevelDB キャッシュストアも廃止され、RocksDB に置き換えられました。

完全なリストについては、[Data Grid 7.3 で非推奨になった特徴と機能](#) を参照してください。



## パート V. DATA GRID サーバーへのパッチ適用

Red Hat Data Grid サーバーは JBoss Enterprise Application Platform (EAP) のパッチ機能を使用するため、既存のインストールを完全に置き換えずに、エラータリリリースからの変更を適用できます。

パッチは、リリースバージョン内の累積的な更新のために配布されます。このリリースの基本バージョンは 7.3.1 です。技術的な問題により、Data Grid 7.3.0 にパッチを適用することはできません。

7.3.x パッチは、基本バージョンまたは他のパッチの上に適用できます。

7.3.x パッチを他の Data Grid リリースバージョンに適用することはできません。同様に、他のリリースバージョンのパッチを 7.3 リリースに適用することはできません。

Data Grid は、サーバーインスタンスのみにパッチを提供します (リモートクライアントサーバーモード)。EAP モジュール、クライアント、Data Grid ライブラリーモードなど、他のすべてのディストリビューションは、フルリリースとして提供されます。

## 第20章 RED HAT DATA GRID へのパッチの適用



### 重要

サーバーモードでは、7.3.0 の上で Data Grid 7.3.1 以降にパッチを適用することはできません。

Data Grid 7.3.1 サーバーは、完全なディストリビューションとして提供されます。Data Grid 7.3.2 サーバー、および後続の各 7.3.x バージョンは、パッチとして提供されます。

Data Grid にパッチを適用するには、次の手順を実行します。

1. Red Hat カスタマーポータル (<https://access.redhat.com/downloads/>) からパッチをダウンロードします。
2. パッチを適用するサーバーインスタンスが実行中の場合は、停止します。クラスロードの問題を回避するために、サーバーの実行中は、Data Grid にパッチを適用しないでください。

管理コンソールを使用して、サーバーを停止するか、Data Grid が実行されているターミナルで **Ctrl-C** を入力します。

3. ターミナルを開き、RHDG\_HOME ディレクトリーに移動します。

```
$ cd RHDG_HOME
```

4. 次のように、パッチを適用します。

```
$ bin/cli.sh "patch apply /path/to/jboss-datagrid-7.3.x-server-patch.zip"
```

5. 次のように、**standalone.sh** または **domain.sh** スクリプトを使用して、サーバーを起動します。

```
$ bin/standalone.sh -c clustered.xml
```

### 7.3.8 および 7.3.9 パッチによるエラーの解決



### 重要

Data Grid サーバーのインストールにパッチを適用して、7.3.8 または 7.3.9 にアップグレードすると、パッチの問題により、起動時に、エラーが発生します。パッチを適用した後、次の手順を完了する必要があります。そうしないと、Data Grid を開始できません。

1. サーバー設定ファイルを開いて、編集します。
2. **http-remoting-connector** をリモートリングサブシステムから削除します。

### 前

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0">
  <http-connector name="http-remoting-connector" connector-ref="default" security-realm="ApplicationRealm"/>
</subsystem>
```

```
</subsystem>
```

#### After

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0"/>
```

3. サーバー設定を保存し、閉じます。

詳細は、[RHDG server fail to start after patch 7.3.8 is applied](#) (Red Hat ナレッジベース) を参照してください。

## 第21章 パッチを元に戻す

パッチをロールバックして、Red Hat Data Grid サーバーを以前にインストールしたバージョンに戻すことができます。



### 重要

予期しない動作や望ましくない影響をもたらすパッチを適用した後のみ、パッチをロールバックする必要があります。パッチのロールバックは、一般的なアンインストール機能を目的としたものではありません。

Data Grid パッチを元に戻すには、次の手順を実行します。

1. ロールバックするサーバーインスタンスが実行中の場合は、停止します。  
管理コンソールを使用して、サーバーを停止するか、Data Grid が実行されているターミナルで **Ctrl-C** を入力します。

2. ターミナルを開き、**RHDG\_HOME** ディレクトリーに移動します。

```
$ cd RHDG_HOME
```

3. ロールバックするパッチの ID を見つけます。

```
$ bin/cli.sh "patch history"
```

4. サーバーのバージョンを次のようにロールバックします。

```
$ bin/cli.sh "patch rollback --patch-id=PATCH_ID --reset-configuration=false"
```



### 警告

**reset-configuration** オプションを指定する際は、注意してください。

**--reset-configuration=false** は、サーバー設定を元に戻しません。パッチを適用すると、サーバー設定が変更される可能性があるため、パッチをロールバックしても、設定をロールバックしないと、サーバーが再起動しない可能性があります。この場合は、サーバーを起動する前に、サーバーの設定を確認し、必要に応じて、手動で調整する必要があります。

**--reset-configuration=true** は、サーバー設定をパッチ適用前の状態に戻します。パッチの適用後、サーバー設定の変更は、すべて削除されます。

パッチをロールバックしようとした際、競合が存在すると、操作は失敗し、警告が発生します。**patch --help** と入力して、競合を解決するために使用できる引数を一覧表示します。

5. **standalone.sh** または **domain.sh** スクリプトを使用して、サーバーを起動します。

