



Red Hat Data Grid 8.2

Red Hat Data Grid 8.2 リリースノート

Data Grid 8.2 のリリース情報を取得する

Red Hat Data Grid 8.2 Red Hat Data Grid 8.2 リリースノート

Data Grid 8.2 のリリース情報を取得する

法律上の通知

Copyright © 2023 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[®] 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 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

Data Grid 8.2 の機能および拡張機能について確認し、現在の既知の問題および解決された問題について確認し、Red Hat がサポートする設定を確認してください。

目次

RED HAT DATA GRID	3
DATA GRID のドキュメント	4
DATA GRID のダウンロード	5
多様性を受け入れるオープンソースの強化	6
第1章 最新の DATA GRID バージョンへのアップグレード	7
第2章 DATA GRID のリリース情報	8
2.1. DATA GRID 8.2.0 の新機能	8
2.2. DATA GRID 8.2.1 の新機能	20
2.3. DATA GRID 8.2.2 の新機能	22
2.4. DATA GRID 8.2.3 の新機能	22
2.5. DATA GRID 8.2 でサポートされる JAVA バージョン	23
第3章 既知の問題および修正された問題	24
3.1. DATA GRID の既知の問題	24
3.2. DATA GRID 8.2.0 で修正された問題	26
3.3. DATA GRID 8.2.1 で修正された問題	27
3.4. DATA GRID 8.2.2 で修正された機能	27
3.5. DATA GRID 8.2.3 で修正された機能	27
3.6. ホストシステムおよび依存関係の問題	28
第4章 テクノロジープレビュー	29
4.1. テクノロジープレビュー機能	29

RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

エラスティックスケーリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- [Data Grid 8.2 ドキュメント](#)
- [Data Grid 8.2 コンポーネントの詳細](#)
- [Data Grid 8.2 でサポートされる設定](#)
- [Data Grid 8 機能のサポート](#)
- [Data Grid で非推奨の機能](#)

DATA GRID のダウンロード

Red Hat カスタマーポータルで [Data Grid Software Downloads](#) にアクセスします。



注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#)をご覧ください。

第1章 最新の DATA GRID バージョンへのアップグレード

Red Hat は、デプロイメントを 8.2.x から最新の Data Grid 8 バージョンにできるだけ早くアップグレードすることをお勧めします。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、[Red Hat Data Grid 製品ドキュメント](#) にあります。

第2章 DATA GRID のリリース情報

新機能および最新の Data Grid のリリース情報についてご確認ください。

2.1. DATA GRID 8.2.0 の新機能

Data Grid 8.2 は、使いやすさ、パフォーマンス、セキュリティーを向上します。新機能を確認してください。

2.1.1. Data Grid Server

Data Grid Server は、Java 仮想マシン (JVM) 向けの柔軟性、耐久性、拡張性の高いデータストアを提供します。

デフォルトの認可

Data Grid Server の設定で認可が有効になり、デフォルトのロールおよびパーミッションのセットに基づいてユーザーアクセスが制限できます。

たとえば、Data Grid クラスターで管理操作の実行を許可する **admin** ロールをユーザーに割り当てる必要があります。サーバーリソースの操作を行うのに十分なパーミッションがないユーザーには、以下のメッセージがログに記録されます。

```
The user is not allowed to access the server resource: ISPN000287: Unauthorized access: subject 'Subject with principal(s): [myusername]' lacks 'ADMIN' permission
```

以下は、認可が有効なデフォルトの Data Grid Server 設定を示しています。

```
<cache-container name="default" statistics="true">
  <transport cluster="${infinispan.cluster.name:cluster}"
    stack="${infinispan.cluster.stack:tcp}"
    node-name="${infinispan.node.name:}"/>
  <security>
    <authorization/> 1
  </security>
</cache-container>
```

- 1 サーバー管理および運用に対する認可を有効にします。 **authorization** 要素を削除して、アクセスを無制限に許可できます。



注記

この設定はキャッシュ設定には影響がありません。キャッシュの認可を個別に有効にする必要があります。

- [ユーザーの作成と変更](#)
- [ユーザーのロールとパーミッション](#)
- [ユーザー認証の設定](#)

セキュリティーを強化するためのクレデンシャルストア

パスワードなどの機密テキストの文字列を保護するには、Data Grid Server の設定を直接操作するのではなく、認証情報キーストアに追加できるようになりました。

Data Grid CLI で **credentials** コマンドを使用して、設定を行います。

- [キーストアへの認証情報の保存](#)

監査ロギング

監査ログを使用すると、Data Grid クラスターへの変更を追跡できるため、変更のタイミングや変更を加えたユーザーを把握できます。

org.infinispan.AUDIT ロギングカテゴリーを使用して、監査ロギングを有効にし、設定イベントおよび管理操作の記録方法を設定します。

- [監査ログ](#)

エンドポイント IP フィルタリング

以下の例のように、Data Grid Server エンドポイントの IP アドレスフィルタリングルールを設定して、接続を許可または拒否できるようになりました。

```
<endpoints socket-binding="default" security-realm="default">
  <ip-filter>
    <accept from="192.168.0.0/16"/> ①
    <accept from="10.0.0.0/8"/> ②
    <reject from="0"/> ③
  </ip-filter>
  ...
</endpoints>
```

- ① **192.168.0.0/16** ブロックの IP アドレスが割り当てられたクライアントからの接続を受け入れます。
- ② **10.0.0.0/8** ブロックの IP アドレスが割り当てられたクライアントからの接続を受け入れます。
- ③ 他の IP アドレスが割り当てられたクライアントからの接続を拒否します。

Data Grid CLI で **server connector ipfilter** コマンドを使用して、IP フィルタールールを検査して変更します。

- [エンドポイントの IP フィルタリング](#)

クライアント証明書検証のトラストストア

Data Grid Server では、サーバーとクライアント間の相互 SSL/TLS 証明書を強制的に検証する必要がある場合に、クライアントトラストストアを **server-identities** 設定に追加できるようになりました。

```
<security-realm name="default">
  <server-identities>
    <ssl>
      <keystore path="server.pfx"
        keystore-password="password" alias="server"/>
      <truststore path="trust.pfx" ①
        relative-to="infinispan.server.config.path"
        password="secret"/>
    </ssl>
```

```

</server-identities>
<truststore-realm/> ②
</security-realm>

```

- ① クライアントアイデンティティーの検証に Data Grid Server が使用するトラストストアを指定します。
- ② すべてのクライアントにパブリック証明書を含めるには、トラストストアが必要です。**truststore-realm** 要素を含めない場合には、トラストストアに必要なのは証明書チェーンのみです。

- [トラストストアレルム](#)

セキュリティーレルムアイデンティティーキャッシング

Data Grid Server は、パフォーマンス向上のために、セキュリティーレルムのアイデンティティーをキャッシュするようになりました。

以下の例のように、**cache-max-size** 属性および **cache-lifespan** 属性でセキュリティーレルムのアイデンティティーキャッシュを設定できます。以下に例を示します。

```

<security-realm name=" cache-max-size="256" ①
    cache-lifespan="-1"> ②
...
</security-realm>

```

- ① アイデンティティーキャッシュの最大サイズを設定します。
- ② アイデンティティーキャッシュ内のエントリーの有効期限を設定します。デフォルトでは、エントリーには有効期限はありません。

- [サーバー設定スキーマ](#)

シングルポートエンドポイントの暗黙的なコネクター

エンドポイントに単一のポートを使用する場合に、Data Grid Server 設定で Hot Rod および REST コネクターを定義する必要がなくなりました。

たとえば、Data Grid 8.2 の場合に、以下の **エンドポイント** 設定は暗黙的にデフォルトの Hot Rod および REST コネクターを使用します。

```

<endpoints socket-binding="default" security-realm="default"/>

```

これとは対照的に、以下の **エンドポイント** 設定には、Hot Rod および REST コネクターを明示的に含めています。

```

<endpoints socket-binding="default" security-realm="default">
  <hotrod-connector name="hotrod"/>
  <rest-connector name="rest"/>
</endpoints>

```

以前のバージョンに合ったメトリクスエンドポイント

バージョン間の互換性を確保するために、Data Grid Server は Data Grid 7.3 以降からのすべてのメトリクスをエクスポートするようになりました。

2.1.2. Data Grid コンソール

Data Grid コンソールには、グラフィカルユーザーインターフェイスが提供されており、リモート Data Grid クラスターを監視および維持できます。

ユーザーエクスペリエンスの向上

Data Grid 8.2 のコンソールでは、ユーザーエクスペリエンスおよびユーザービリティが複数強化されています。

- ユーザーが存在しない場合には、ウェルカムページからユーザーを作成するように求めらる。
- [PatternFly UX の記述ガイドライン](#) に合わせて、すべてのテキスト文字列がレビューおよび編集されてた。
- フォームやラベルや説明内のコンテキストヘルプが明確になるように更新された。

ロールベースのアクセス制御

Data Grid コンソールは、セキュリティ認可の設定を適用し、割り当てられたロールおよびパーミッションに基づいてユーザーアクセスを制限します。

カウンター管理

本リリースで Data Grid コンソールのカウンターの管理が改善され、カウンターの削除およびフィルターができるようになりました。

2.1.3. Data Grid コマンドラインインターフェイス

Data Grid コマンドラインインターフェイス (CLI) を使用すると、リモート Data Grid クラスターで管理操作を実行できます。

Data Grid クラスターのバックアップおよび復元

CLI には、**backup** コマンドが含まれており、キャッシュされたエントリー、キャッシュ設定、Protobuf スキーマ、およびサーバースクリプトを含む Data Grid リソースのアーカイブを作成できます。のちほど、再起動または移行後に、バックアップアーカイブから Data Grid クラスターを復元できます。

- [Data Grid クラスターのバックアップおよび復元](#)
- [backup コマンドリファレンス](#)

パフォーマンステストツール

benchmark コマンドを使用すると、CLI を使用してキャッシュに対してパフォーマンステストを実行できます。

- [Benchmark コマンドリファレンス](#)

ユーザーロールとパーミッションの割り当て

CLI の **user** コマンドで **roles** サブコマンドが拡張され、ユーザーロールを表示、付与、拒否できるようになりました。ユーザーのロール割り当てを動的に更新して、認可設定を制御し、Data Grid クラスターおよびキャッシュへのアクセスを制限できるようになりました。

- [ユーザーコマンドリファレンス](#)

クロスサイトレプリケーション操作

本リリースでは、CLI の **site** コマンドを使用して、追加のクロスサイトレプリケーション操作を実施できるようになりました。

site name: ローカルサイトの名前を返します。

site view: 相互にバックアップ可能な全サイトの名前一覧を返します。

site state-transfer-mode: クロスサイトの状態遷移を手動または自動的に行うように設定します。

- [Site コマンドリファレンス](#)

クレデンシャルキーストアの管理

CLI には、**credentials** コマンドが含まれており、Data Grid Server の認証情報キーストアを管理できます。

- [Credentials コマンドリファレンス](#)

ネイティブ CLI

Data Grid 8.2 は、Linux、macOS、または Windows で実行できるネイティブ CLI を追加し、**oc** クライアントプラグインとして使用できます。

1. Red Hat カスタマーポータル [Data Grid Software Downloads](#) からネイティブ CLI をダウンロードします。
2. インストール手順と使用例については、ディストリビューションに含まれている **README** を開いてください。



注記

ネイティブ CLI は現在、[テクノロジープレビュー機能](#) として利用できます。

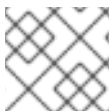
2.1.4. クロスサイトレプリケーション

クロスサイトレプリケーションを使用すると、複数の地理的リージョンにまたがる Data Grid クラスターをバックアップできます。

状態遷移の自動操作

問題が発生し、バックアップの場所がオフラインになった場合に、CLI、JMX または REST 経由で、クロスサイトの状態遷移操作を手動で実行する必要があります。

ただし、非同期バックアップストラテジーを使用する場合には、Data Grid は競合の解決後に、クロスサイトの状態遷移操作を自動的に実行できるようになりました。バックアップの場所がオンラインになり、ネットワーク接続が安定していることを検知すると、Data Grid はバックアップの場所の間で双方向状態遷移を開始します。たとえば、Data Grid は、状態を **LON** から **NYC** に、**NYC** から **LON** に同時に遷移します。



注記

自動状態遷移は、非同期バックアップストラテジーでのみ可能です。

- [自動状態遷移](#)

カスタム競合解決の SPI

Data Grid には SPI が含まれており、非同期 Active/Active バックアップ設定の競合解決をカスタマイズできます。

XSiteMergePolicy 列挙は、以下の競合解決のオプションを提供します。

DEFAULT

同時書き込みからの競合を処理するには、デフォルトのアルゴリズムを使用します。

PREFER_NON_NULL

書き込み/削除の競合が発生した場合には、このアルゴリズムは書き込み操作を維持し、削除操作を破棄します。デフォルトのアルゴリズムは他のすべての競合に適用されます。

PREFER_NULL

書き込み/削除の競合が発生した場合には、このアルゴリズムは削除操作を維持し、書き込み操作を破棄します。デフォルトのアルゴリズムは他のすべての競合に適用されます。

ALWAYS_REMOVE

両方のサイトから競合するエントリーを削除します。

以下のような **merge-policy** 属性を使用して、カスタム実装を含む競合解決ポリシーを指定できます。

```
<distributed-cache name="eu-customers">
  <backups merge-policy="org.mycompany.MyCustomXSiteEntryMergePolicy">
    <backup site="LON" strategy="ASYNC"/>
  </backups>
</distributed-cache>
```

- [クロスサイトの競合解決の設定](#)
- [org.infinispan.xsite.spi.XSiteEntryMergePolicy](#)

CLI および REST からクロスサイトビューを検証する機能

CLI または REST API でクロスサイトビューを検証できるようになりました。

CLI の場合は、**site view** コマンドを実行して、相互にバックアップするすべてのサイトの名前の一覧を取得します。

REST API から、以下の **GET** 要求を呼び出します。

```
GET /rest/v2/cache-managers/{cacheManagerName}
```

以下のように、Data Grid は、JSON 形式のバックアップ場所のリストで応答します。

```
"sites_view": [
  "LON",
  "NYC"
]
```

- [Site コマンドリファレンス](#)
- [Basic Cache Manager 情報の取得 \(REST API\)](#)

2.1.5. Hot Rod クライアント

Hot Rod は、カスタムバイナリー TCP プロトコルで、異なるプログラミング言語で、クライアントアプリケーションに高パフォーマンスでデータアクセスできるようにします。



重要

Hot Rod クライアントで Java 8 を使用している場合は、Data Grid 8.2 で致命的な **SSLHandshakeException** エラーを回避するために、少なくとも Java 8u252 にアップグレードする必要があります。詳細は、[既知の問題](#) を参照してください。

ニアキャッシュパフォーマンスの改善

Data Grid Server には、bloom フィルターが追加され、インバリデーションメッセージの合計数を減らして、書き込み操作のパフォーマンスが最適化されるようになりました。

`nearCacheUseBloomFilter()` メソッドでニアキャッシュの bloom フィルターを有効にします。

- [ニアキャッシュ](#)

新規の Hot Rod クライアント設定プロパティ

Data Grid 8.2 以降、Hot Rod クライアント設定 API には、以下の設定プロパティが含まれるようになりました。

- `infinispan.client.hotrod.transport_factory` は使用するトランスポートファクトリーを指定します。デフォルトは `org.infinispan.client.hotrod.impl.transport.netty.DefaultTransportFactory` です。
- `infinispan.client.hotrod.cache.<cache_name>.marshaller` は、キャッシュごとに使用するマーシャラーを指定します。
- `infinispan.client.hotrod.ssl_ciphers` は、優先順にスペース区切りの暗号化をリストし、SSL ハンドシェイク時に使用して、鍵暗号化の暗号アルゴリズムをネゴシエートします。
- `infinispan.client.hotrod.ssl_provider` は、SSL エンジンの作成時に使用するセキュリティプロバイダーを指定し、OpenSSL にデフォルト設定します。
- `infinispan.client.hotrod.cache.<cache_name>.transaction.transaction_manager_lookup` は、キャッシュごとに使用する `TransactionManagerLookup` を指定します。



注記

`infinispan.client.hotrod.trust_store_path` などの一部のプロパティが非推奨になりました。詳細は、Red Hat ナレッジベースの [Deprecation and removals](#) の記事を参照してください。

- [Hot Rod クライアント設定 API](#)

2.1.6. クエリー API

Data Grid Query API を使用すると、リレーショナルまたは完全テキストクエリーを使用して、Ickle クエリー言語でキャッシュと検索値をインデックス化できます。

Data Grid 8.2 では、Hibernate Search 6 をベースにすることでクエリー実装が大幅に改善され、Apache Lucene 8 インデックス機能をサポートするようになりました。本リリースでは、以下のクエリー拡張機能を提供します。

- 高速なインデックス。
- インデックス付き、インデックスなし、およびハイブリッドクエリーの統計。
- 厳密に型指定されたインデックス設定。これは、文字列キー/値のプロパティを置き換えます。

インデックス作成とクエリーに関する包括的なドキュメントについては、[キャッシュ内の値のクエリー](#)を参照してください。



注記

移行の詳細を確認し、Data Grid 8.2 のクエリー設定をどのように適応させるかを確認します。

詳細は、[データグリッド移行ガイド](#)を参照してください。

2.1.7. REST API

Data Grid REST API を使用すると、リモートクラスターと対話し、HTTP 経由でキャッシュできます。

鍵とエントリーのストリーミング

Data Grid REST API では、キャッシュ内のすべてのキーまたはエントリーを JSON 形式で取得できるようになりました。以下のように **GET** 要求を呼び出します。

キーのストリーミング

```
GET /rest/v2/caches/{cacheName}?action=keys
```

エントリーのストリーミング

```
GET /rest/v2/caches/{cacheName}?action=entries
```

- [キャッシュからのすべてのキーの取得](#)
- [キャッシュからのすべてのエントリーの取得](#)

アクセス制御リストキャッシュの使用

Data Grid 8.2 には、ユーザーのロールマッピングを格納するアクセス制御リスト (ACL) キャッシュが含まれます。ACL キャッシュは、REST API 経由で対話できます。

ユーザー ACL 情報の表示

```
GET /rest/v2/security/user/acl
```

ACL キャッシュのフラッシュ

```
POST /rest/v2/security/cache?action=flush
```

- [ユーザーの ACL の取得](#)

クエリーとインデックスの統計取得

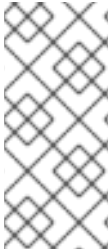
GET 要求を使用して、キャッシュでクエリーとインデックスに関する情報を取得します。

```
GET /v2/caches/{cacheName}/search/stats
```

- [クエリーおよびインデックス統計の取得](#)

キャッシュ設定

Data Grid REST API では、キャッシュの設定取得時の応答が改善され、リモートクラスターのキャッシュ設定とローカルプロジェクトのキャッシュ設定の比較および検証が容易になりました。



注記

キャッシュ設定に非推奨の属性が含まれる場合には、Data Grid により、現在のスキーマと互換性を確保するため、その属性は自動的に変換されます。

キャッシュ設定を簡単に比較できるように、アプリケーションで常に最新のスキーマを使用する必要があります。

2.1.8. Data Grid マーシャリング

Data Grid には、他のマーシャラー実装に加えて、ProtoStream API が追加され、ネットワーク全体および永続ストレージにカスタム Java オブジェクトを送信できるようになりました。

ProtoStream

Data Grid 8.2 は、ProtoStream API を 4.4.1.Final にアップグレードします。



注記

Data Grid 8.2 の ProtoStream API を変更すると、以前の Data Grid 8 バージョンからのアップグレードに影響します。

詳細は、[Data Grid 8 のアップグレードに関する注意事項](#) を参照してください。

デシリアイズの許可リスト

Red Hat では、多様性を受け入れる用語の使用への取り組みに努めており、Java クラスのシリアル化を設定する際に使用するホワイトリストという用語を許可リストに変更しています。

Data Grid 8.1

```
<cache-container>
  <serialization>
    <white-list>
      <class>org.infinispan.test.data.Person</class>
      <regex>org.infinispan.test.data.*</regex>
    </white-list>
  </serialization>
</cache-container>
```

Data Grid 8.2

```
<cache-container>
  <serialization>
    <allow-list>
      <class>org.infinispan.test.data.Person</class>
      <regex>org.infinispan.test.data.*</regex>
    </allow-list>
  </serialization>
</cache-container>
```

SerializationContextInitializer 実装の作成

Data Grid には、**@AutoProtoSchemaBuilder** アノテーションが追加され、**SerializationContextInitializer** を拡張するクラスまたはインターフェイスの実装を生成できるようになりました。これにより、Data Grid キャッシュにカスタム Java オブジェクトを保存するときに Protobuf スキーマおよびマーシャラーを作成するメカニズムの効率性と信頼性が強化されます。



注記

以前の Data Grid バージョンでは、**MessageMarshaller** API および **ProtoSchemaBuilder** アノテーションを使用して Protobuf スキーマを作成していました。**@AutoProtoSchemaBuilder** アノテーションに移行して、これの使用を開始する必要があります。

- [ProtoStream アノテーション](#)
- [シリアル化のコンテキストイニシャライザーの作成](#)
- [アプリケーションの AutoProtoSchemaBuilder アノテーションへの移行](#)

外部クラスのマーシャリング用の @ProtoAdaptor

Data Grid では **@ProtoAdaptor** アノテーションのサポートが追加され、外部のサードパーティー Java オブジェクトクラスのアダプタークラスに追加できます。

- [ProtoStream アダプタークラスの作成](#)

コレクションや配列の値としての直接使用

Data Grid 8.2 では、**Array List**、**Linked List**、**Hash Set**、**Linked Hash Set**、**Tree Set**、および **String[]** や **int[]** のような単純な型の配列の値を Proto Stream API で使用できます。

以前のバージョンの Data Grid では、マッパーを追加せずにコレクションや配列を直接、値として使用できませんでした。**put (... <Array List>)** を呼び出すと、次のような例外が発生します。

```
IllegalArgumentExcpion: No marshaller registered for Java type java.util.ArrayList
```

Kyro および Protostuff marshallers が非推奨に

Kyro および Protostuff マーシャラーが非推奨になりました。詳細は、[Red Hat ナレッジベースの非推奨と削除](#)の記事を参照してください。

2.1.9. Data Grid の設定

Data Grid では、キャッシュ用だけでなく、セキュリティーやクラスタートランスポートなどの基盤となるメカニズムをカスタマイズできるように、スキーマベースの設定オプションを提供します。

認可: ユーザーロールおよびパーミッション

Data Grid 8.2 では、Data Grid インストールへのアクセスとキャッシュへのアクセスのセキュリティーを確保するロールベースアクセス制御 (RBAC) 機能が改善されました。

Cache Manager アクセスの認可を有効にするには、以下の例のように **authorization** 要素を **cache-container** に追加します。

```
<cache-container name="default" statistics="true">
  <security>
    <authorization/>
  </security>
</cache-container>
```

キャッシュの認可を有効にするには、以下のように **authorization** 要素を追加します。

```
<distributed-cache name="myCache" mode="SYNC">
  <security>
    <authorization/>
```

</security>
</distributed-cache>

クラスターのロールマッパー

Data Grid 8.2 では、**ClusterRoleMapper** が導入されました。これは、Data Grid がセキュリティープリンシパルを承認ロールに関連付けるために使用するデフォルトのメカニズムです。

このロールマッパーは永続レプリケートされたキャッシュを使用して、デフォルトのロールおよびパーミッションのプリンシパルからロールへのマッピングを動的に保存します。

表2.1 デフォルトのユーザーロール

ロール	パーミッション	説明
admin	ALL	Cache Manager ライフサイクルの制御など、すべてのパーミッションを持つスーパーユーザー。
deployer	ALL_READ、ALL_WRITE、LISTEN、EXEC、MONITOR、CREATE	application パーミッションに加えて、Data Grid リソースを作成および削除できます。
application	ALL_READ、ALL_WRITE、LISTEN、EXEC、MONITOR	observer パーミッションに加え、Data Grid リソースへの読み取りおよび書き込みアクセスがあります。また、イベントをリスンし、サーバータスクおよびスクリプトを実行することもできます。
observer	ALL_READ、MONITOR	monitor パーミッションに加え、Data Grid リソースへの読み取りアクセスがあります。
monitor	MONITOR	JMX および metrics エンドポイント経由で統計を表示できます。

新規パーミッション

CREATE パーミッションを使用すると、ユーザーはキャッシュ、カウンター、スキーマ、スクリプトなどのコンテナリソースを作成および削除することができます。



注記

CREATE パーミッションは、**__schema_manager** および **__script_manager** ロールを置き換えます。このロールは、Data Grid Server に対してスキーマおよびスクリプトを追加および削除するのに必要です。

MONITOR パーミッションは、JMX 統計および **metrics** エンドポイントへのアクセスを許可します。

承認の詳細は、**セキュリティーガイド**の以下の内容を参照してください。

- [ロールマッパー](#)
- [アクセス制御リスト \(ACL\) キャッシュ](#)
- [ユーザーのロールとパーミッション](#)
- [パーミッション](#)

Data Grid キャッシュ設定のフラグメント

Data Grid 8.2 では、キャッシュ設定に **infinispan** および **cache-container** 要素を含める必要がなくなりました。

キャッシュの作成時に指定する必要があるのは、***-cache** 要素のみです。

たとえば、同期モードを使用する分散キャッシュを作成するには、以下の設定を使用できます。

```
<distributed-cache name="myCache" mode="SYNC" />
```

エントリーの Protobuf エンコーディングを使用するレプリケートキャッシュを作成するには、以下の設定を使用できます。

```
<replicated-cache name="books">
  <encoding media-type="application/x-protostream"/>
</replicated-cache>
```

JGroups INSERT_BEFORE 属性

INSERT_BEFORE 値を使用して、**stack.combine** 継承属性で JGroups クラスタートランSPORTをカスタマイズできます。

```
<ASYM_ENCRYPT asym_keylength="2048"
  asym_algorithm="RSA"
  change_key_on_coord_leave = "false"
  change_key_on_leave = "false"
  use_external_key_exchange = "true"
  stack.combine="INSERT_BEFORE" ❶
  stack.position="pbcast.NAKACK2"/>
```

- ❶ JGroups スタックの **pbcast.NAKACK2** の前に **ASYM_ENCRYPT** プロトコルを挿入します。



注記

Data Grid 8.2 スキーマには、**INSERT_ABOVE** および **INSERT_BELOW** 属性も含まれています。

INSERT_ABOVE は **INSERT_AFTER** と同じです。 **INSERT_BELOW** は **INSERT_BEFORE** と同じです。

- [継承属性](#)

JGroups デフォルトスタック

Data Grid 8.2 では、デフォルトの JGroups スタックにおける UNICAST3 および NAKACK2 プロトコルの再送信要求設定が変更されました。

ガベージコレクション (GC) の一時停止が長い場合など、ノードはクラスター内の他のノードからの JGroups メッセージを処理できない場合があります。これらのノードが再び利用可能な状態になったら、XMIT 要求を使用して送信ノードを再転送するように要求します。

再送信要求の失敗からのクラスター転送の問題を回避するために、以下の変更が適用されます。

- **xmit_interval** プロパティの値が 100 ミリ秒から 200 ミリ秒に増える。
- **max_xmit_req_size** プロパティは、UDP の最大値が 8500、TCP が 64000 ではなく、再送信要求の最大メッセージ数を 500 に設定するようになりました。

キャッシュのヘルス

Data Grid には、新しい **FAILED** ステータスが追加されています。このバージョンで、キャッシュに利用可能なヘルスステータスは以下の通りです。

ヘルスステータス	説明
HEALTHY	キャッシュが想定どおりに動作していることを示します。
HEALTHY_REBALANCING	キャッシュがリバランス状態にあることを示しますが、それ以外場合は想定どおりに動作します。
DEGRADED	キャッシュが想定どおりに動作しておらず、トラブルシューティングが必要になる場合があることを示します。
FAILED	8.2 で追加され、キャッシュが指定の設定で起動できなかったことを示します。

JDBC 文字列ベースのキャッシュストア

JDBC 文字列ベースのキャッシュストアでは、キャッシュエントリの保存に使用されるデータテーブルに加えて **a_META** テーブルが作成されます。**The META** テーブルには、メタデータがあり、このメタデータにより、既存のデータベースコンテンツが現在の Data Grid バージョンおよび設定と互換性を確保します。

2.1.10. Spring アプリケーション

Data Grid は、Spring キャッシュと Spring セッション実装を提供します。

Data Grid 8.2 の時点で、Spring アプリケーションは ProtoStream マーシャラーを使用して、Java オブジェクトを Protocol Buffers (Protobuf) 形式でエンコードおよびデコードできます。

詳細は、[Spring Boot スターター](#) を参照してください。

2.2. DATA GRID 8.2.1 の新機能

Data Grid 8.2.1 の新機能を参照してください。

2.2.1. Data Grid Server

本リリースには、Data Grid Server の機能拡張がいくつか含まれています。

データソース接続

Data Grid Server により、JDBC キャッシュストアなどのデータソースとの無効な接続の検出および管理が容易になりました。

background-validation 属性および **validate-on-acquisition** 属性は接続プールプロパティに含まれます。Data Grid コマンドラインインターフェイス (CLI) には、データソース接続の一覧表示とテストを可能にする **server datasource** コマンドが含まれています。

詳細は以下を参照してください。

- [データソースのテスト](#)
- [JDBC キャッシュストアのデータソース設定](#)
- [CLI Command Reference: Server](#)

LDAP アイデンティティの再帰検索

search-recursive="true" パラメーターが LDAP レルムで使用可能になり、再帰検索が可能になりました。詳細は、[LDAP Realms](#) を参照してください。

TLSv1.3 サポート

Data Grid Server は、デフォルトで TLS バージョン 1.2 および 1.3 をサポートします。

TLS 1.3 のみを許可する場合は、Data Grid Server が使用する TLS バージョンを設定できます。詳細は、[TLS バージョンおよび暗号スイートの設定](#) を参照してください。

2.2.2. Data Grid コンソール

Data Grid Console は、キャッシュ用の **Entries** タブを表示しなくなったり、エンコーディングを設定していないキャッシュのエントリを作成できるようになりました。

Data Grid は、コンソールでエントリを作成または変更する場合は、**application/x-protostream** メディアタイプでキャッシュエンコーディングを設定することを推奨します。詳細は、[Cache Encoding and Marshalling](#) を参照してください。

2.2.3. Hot Rod Node.JS クライアント

Hot Rod Node.JS クライアントは、**PLAIN** に加えて **DIGEST-MD5** および **SCRAM** 認証メカニズムに対応するようになりました。

さまざまな SASL 認証メカニズムを設定する方法と、[Hot Rod Node.JS Client Guide](#) で使用例を取得する方法を説明します。

2.2.4. ドキュメント

本リリースでは、ドキュメントに関する主な改善およびリビジョンについて以下を行います。

- Data Grid サーバーガイドの [Encrypting Data Grid Server Connections](#) に記載されている、リモートクライアント接続を保護するために Java キーストアとトラストストアを追加する手順を更新しました。
- JMX 管理用にリモートポートを有効にする情報を、[Embedding Data Grid: Enabling JMX Remote Ports – Data Grid Server: Enabling JMX Remote Ports](#) に追加しました。

2.3. DATA GRID 8.2.2 の新機能

Data Grid 8.2.2 の新機能を参照してください。

2.3.1. CVE-2021-44228 のセキュリティーパッチ

Data Grid 8.2.2 は、Apache Log4j ロギングライブラリーのセキュリティー脆弱性である [CVE-2021-44228](#) を修正します。Data Grid には、Data Grid Server ディストリビューションの一部として、影響を受ける **log4j-core** ライブラリーと、Red Hat OpenShift デプロイメントの Data Grid Server イメージが含まれます。

Red Hat は、できるだけ早く 8.2.2 にアップグレードすることを推奨します。アップグレードできない場合は、[RHSB-2021-009 Log4Shell - Remote Code Execution](#) security bulletin でこの脆弱性を軽減する手順に従うことを推奨します。

また、Red Hat は以下を推奨しています。

- Hot Rod クライアントや Data Grid が組み込みライブラリーとして含まれるプロジェクトで、**log4j** の依存関係のバージョンを確認します。
- Red Hat JBoss EAP のデプロイメントをチェックして、EAP 用の Data Grid モジュールが対象の **log4j** の依存関係を含んでいない場合でも、この脆弱性による影響がないかを確認してください。

この脆弱性の Data Grid への影響については、Red Hat ナレッジベースの [Is Red Hat Data Grid 7.x/8.x impacted by CVE-2021-44228 または CVE-2021-4104?](#) を参照してください。

2.3.2. 最大アイドル有効期限

Data Grid は、touch コマンドを送信して、クラスター間での最大アイドル時間、**max-idle** の有効期限に対するタイムアウト値を調整します。有効期限設定の **touch** 属性を使用して、タッチコマンドを同期または非同期に送信するように Data Grid を設定できます。詳細は、[Data Grid 設定スキーマのリファレンス](#) を参照してください。

2.3.3. クロスサイトレプリケーション

オフラインのサイトを扱う際のパフォーマンス向上

8.2.2 以降のデータグリッドでは、バックアップロケーションがオフラインになったときのイベントを単一のスレッドで処理しています。

非同期の競合をによる Tombstone リーク

Data Grid 8.2.2 では、非同期バックアップ戦略によるクロスサイトレプリケーションで Tombstone リークが発生する問題が解決されています。Tombstone は、同時書き込みによる競合を解決するためのメカニズムの一部として、リモートサイトが保存していた削除済みのオブジェクトです。

2.4. DATA GRID 8.2.3 の新機能

Data Grid 8.2.3 の新機能を参照してください。

2.4.1. Apache Log4j ライブラリーの脆弱性に対するセキュリティーパッチ

Data Grid 8.2.3 では、Apache Log4j ロギングライブラリーの以下の CVE (Common Vulnerabilities and Exposures) が修正されています。

- [CVE-2021-44832](#) JNDI URI を参照するデータソースが含まれる JDBC アペンダーによるリモートコードの実行
- [CVE-2021-45046](#) スレッドコンテキストメッセージパターンとコンテキストルックアップパターンを使用した log4j2.x の DoS (Denial of Service) 攻撃
- [CVE-2021-45105](#) Thread Context Map (MDC) 入力データを使用した log4j 2.x の DoS (Denial of Service) には、再帰ルックアップとコンテキストルックアップパターンが含まれています

Data Grid には、Data Grid Server ディストリビューションの一部として、影響を受ける log4j ライブラリーと、Red Hat OpenShift デプロイメントの Data Grid Server イメージが含まれます。

Red Hat は、できるだけ早く 8.2.3 にアップグレードすることを推奨します。アップグレードできない場合は、上記の各 Log4j 脆弱性のセキュリティーアドバイザリーページに記載されている軽減策に従うことを Red Hat は推奨します。

また、Red Hat は以下を推奨しています。

- Hot Rod クライアントや Data Grid が組み込みライブラリーとして含まれるプロジェクトで、**log4j** の依存関係のバージョンを確認します。
- Red Hat JBoss EAP のデプロイメントをチェックして、EAP 用の Data Grid モジュールが対象の **log4j** の依存関係を含んでいない場合でも、この脆弱性による影響がないかを確認してください。

2.5. DATA GRID 8.2 でサポートされる JAVA バージョン

Red Hat は、Data Grid のインストール方法に応じて、さまざまな Java バージョンをサポートします。

組み込みキャッシュ

Red Hat は、カスタムアプリケーションでの埋め込みキャッシュに、Data Grid を使用する場合に Java 8 および Java 11 をサポートします。

リモートキャッシュ

Red Hat は、Data Grid Server のインストールに限り、Java 11 をサポートします。Hot Rod Java クライアントの場合には、Red Hat は Java 8 および Java 11 をサポートします。

Java 8 の非推奨

Data Grid 8.2 の時点では、Java 8 のサポートは非推奨になり、Data Grid 8.4 で削除される予定です。

カスタムアプリケーションに埋め込まれたキャッシュを使用する場合は、サポートが利用可能になると Java 11 または Java 17 にアップグレードする計画が必要です。

Java 8 を必要とするアプリケーションで実行している Hot Rod Java クライアントは、古いバージョンのクライアントライブラリーを引き続き使用できます。Red Hat は、最新の Data Grid Server バージョンと組み合わせて、以前の Hot Rod Java クライアントバージョンの使用をサポートしています。

関連情報

- [Data Grid 8.2 でサポートされる設定](#)
- [Data Grid で非推奨の機能](#)

第3章 既知の問題および修正された問題

Data Grid の既知の問題や、修正された問題を確認してください。

3.1. DATA GRID の既知の問題

Red Hat OpenShift で実行している Data Grid クラスターに影響する問題は、[Data Grid Operator 8.2 release notes](#) を参照してください。

Red Hat JBoss EAP (EAP) の Data Grid モジュールに依存関係がない

問題: [JDG-5104](#)

説明: EAP 7.4 の Data Grid モジュールには、必要なすべての Hibernate アーティファクトが含まれていません。さらに、EAP 7.4 は、Data Grid が必要とする Eclipse MicroProfile および SmallRye モジュールを削除します。

回避策: 次の手順を実行します。

1. Data Grid サーバーのディストリビューションをダウンロードします。
2. ターミナルウィンドウを開き、`$RHDG_HOME/server/lib` ディレクトリーに移動します。
3. 次の Hibernate JAR ファイルを見つけます。
 - **hibernate-search-backend-lucene-6.0.2.Final-redhat-00002.jar**
 - **hibernate-commons-annotations-5.0.5.Final-redhat-00002.jar**
4. JAR ファイルを Data Grid モジュールインストールの次のディレクトリーにコピーします。
5. `/modules/system/add-ons/rhdg/org/infinispan/rhdg-8.2/module.xml` を開いて編集します。
6. `optional="true"` 属性を次のモジュールに追加します。

```
modules/system/add-ons/rhdg/org/infinispan/rhdg-8.2/

<module name="org.eclipse.microprofile.config.api" export="true" optional="true"/>
<module name="org.eclipse.microprofile.metrics.api" export="true" optional="true"/>
<module name="io.smallrye.config" services="export" export="true" optional="true"/>
<module name="io.smallrye.metrics" services="import" export="true" optional="true"/>
```

`putAll()` の操作で、楽観的ロックで期限切れのエントリーに書き込むとデッドロックが発生する。

問題: [JDG-5087](#)

説明: 楽観的ロックを使用するトランザクションキャッシュでは、期限切れのエントリーを削除するコマンドが、期限切れが書き込み操作によって引き起こされた場合でもロックを取得してしまいます。`putAll()` 操作で期限切れのエントリーに書き込まれる場合に、この動作によりデッドロックになってしまう場合があります。

回避策: この問題の回避策はありません。

期限切れのエントリーを `putAll()` または `getAll()` で処理するとデータ競合が発生する。

問題: [JDG-5028](#)

説明putAll() または **getAll()** の操作が 2 つ以上のエントリーに影響を与えると、それらのエントリーの両方が期限切れになることがあります。両方のエントリーのキーが同じ **HashMap** バケットにマッピングされている場合には、片方の更新が失われ、Data Grid は次のような例外を出力します。

```
IllegalStateException: Entry should be always wrapped!
```

回避策: この問題の回避策はありません。

クライアントは、TLS/SSL 暗号化を使用するリモートキャッシュに接続できない

問題: [JDG-4763](#)

説明: クライアントはリモートキャッシュに接続できず、Data Grid ログは TLS/SSL に関連する **WARN** ログメッセージを出力します。

この問題は、Data Grid に含まれる WildFly OpenSSL ライブラリーが原因で発生します。ログメッセージの詳細は、Red Hat ナレッジベースのアーティクル [Clients are not able to connect a server after update to RHDG 8.2.1](#) を参照してください。

回避策: 次のプロパティを使用して Data Grid Server を起動し、OpenSSL の代わりに Java TLS/SSL ライブラリーを使用します。

```
-Dorg.infinispan.openssl=false
```

Red Hat JBoss Web Server から Data Grid 8.2 へのセッションの外部化が Tomcat セッションクライアントの 7.3.8 または 8.1.1 バージョンで利用できる

問題: [JDG-4599](#)

説明: Data Grid 8.2 には Tomcat セッションクライアントがまだ含まれていません。これは、EAP 7.4 GA 後に利用できます。

回避策: 以下の設定で、Tomcat セッションクライアントの Data Grid 7.3.8 または 8.1.1 バージョンと組み合わせて、Data Grid Server 8.2 を使用します。

```
<Manager className="org.wildfly.clustering.tomcat.hotrod.HotRodManager"
  configurationName="default"
  persistenceStrategy="${persistenceStrategy}"
  server_list="127.0.0.1:11222"
  protocol_version="2.9"
  auth_realm="default"
  sasl_mechanism="DIGEST-MD5"
  auth_server_name="infinispan"
  auth_username="admin"
  auth_password="changeme"/>
```

Java 8 を使用する Hot Rod クライアントはアップグレードして、SSLHandshakeException を回避する必要がある

問題: [JDG-4279](#)

説明: Hot Rod クライアントで JDK 8 を使用しており、Java のバージョンが少なくとも 8u252 でない場合に以下の致命的な例外になります。

SSLHandshakeException: Remote host closed connection during handshake at
sun.security.ssl.SSLSocketImpl

回避策: 最低でも Java 8u252 を使用していることを確認してください。このバージョンには、Application-Layer Protocol Negotiation (ALPN) に必要なセキュリティ機能が含まれています。

Data Grid 競合解決のパフォーマンス

問題: [JDG-3636](#)

説明: テストケースによっては、Data Grid パーティション処理機能では、競合解決に、想定よりも時間がかかりました。

回避策: この問題の回避策はありません。

Data Grid は JWS セッションを正しくパッシベーションしない

問題: [JDG-2796](#)

説明: JBoss Web Server (JWS) からセッションを外部化する時に **FINE** 永続ストラテジーを使用する場合は、セッションが正しくパッシベートされません。

回避策: この問題の回避策はありません。

3.2. DATA GRID 8.2.0 で修正された問題

Data Grid 8.2.0 には、以下の主な修正が含まれています。

- [JDG-4315](#) トランザクションキャッシュでローリングアップグレードを行うと `IllegalArgumentException` が発生する
- [JDG-3972](#) インデックス化されたキャッシュにインデックス化されていない Protobuf エンティティーが含まれると、クエリーの動作に一貫性がなくなる
- [JDG-4520](#) `DB2TableManager` はテーブルの存在チェック中にすべてのスキーマを確認する
- [JDG-4425](#) `JGroups` 再送信要求の頻度が高く、量が多い
- [eap-4420](#) EAP の埋め込みモジュールで `jboss-marshalling` を使用してカスタムオブジェクトを配置すると `ClassNotFoundException` が発生する
- [JDG-4387](#) 簡単なキャッシュで統計を有効にすると、`EvictionManagerImpl` で `NullPointerException` が発生する
- [JDG-4375](#) TLS 1.1 を無効にして TLS 1.2 だけを有効にする方法
- [JDG-4370](#) EAP モジュールでは AWS の依存関係をオプションにする
- [JDG-4351](#) `IdentityIntMap#clear()` で大きいアレイに対して `Arrays.fill(keys, null)` を呼び出すことの代償として、パフォーマンスが低下し、トランザクションがタイムアウトすることがある。
- [JDG-4344](#) エントリーに大きい値を指定すると HotRod クライアントでメモリーリークが発生する
- [JDG-4339](#) トランザクションキャッシュで競合解決に失敗する

- [JDG-4315](#) トランザクションキャッシュでローリングアップグレードを実行すると `IllegalArgumentException` が発生する
- [JDG-4281](#) Infinispan operator CR はロギングレベルでの変更を受け入れない
- [JDG-4152](#) Infinispan CR は公開設定を削除する際に幂等ではない

3.3. DATA GRID 8.2.1 で修正された問題

Data Grid 8.2.1 には、以下の主な修正が含まれています。

- [JDG-4678](#) キャッシュが Single File キャッシュストアにデータを永続化すると、Data Grid 8.1 から Data Grid 8.2 へのアップグレードに失敗し、データが破損する
- [JDG-4713](#) グローバルな状態の非互換性
- [JDG-4649](#) `CacheContainer.getCache(String cacheName)` でキャッシュを取得する際のスレッドの競合
- [JDG-4590](#) JSON データタイプでキャッシュを照会すると `UnsupportedOperationException` が発生する
- [JDG-4563](#) 暗黙的なキャッシュロックの解除失敗
- [JDG-4438](#) 悲観的なキャッシュでは同時修正が成功
- [JDG-4414](#) 2 つの Pod のいずれかを削除してから 1 つにスケールダウンするとクラスターの状態が破損
- [JDG-3970](#) スロットル TLS ハンドシェイクの失敗 `NotSslRecordException` WARN メッセージ

3.4. DATA GRID 8.2.2 で修正された機能

Data Grid 8.2.2 には、以下の主な修正が含まれています。

- [JDG-5036](#) 攻撃者が制御する文字列の値がログに含まれる場合の Log4j 2.x でのリモートコード実行
- [JDG-4947](#) クロスサイトレプリケーションの tombstone リーク。
- [JDG-4984](#) `WrappedByteArray` のログには値すべてが含まれるため、ログファイルのサイズが不必要に大きくなる。
- [JDG-5043](#) JDBC キャッシュストアを使用したときに結果セットが終了せず、ログファイルの警告が発生する場合があった。

3.5. DATA GRID 8.2.3 で修正された機能

Data Grid 8.2.3 には、以下の主な修正が含まれています。

- [JDG-5060](#) log4j-core: スレッドコンテキストメッセージパターンとコンテキストルックアップパターンを使用した log4j2.x の DoS
- [JDG-5068](#) JDBC アペンダーを介した Log4j 2.x でのリモートコードの実行。

- [JDG-5066](#) log4j-core: Thread Context Map (MDC) 入力データを使用した log4j 2.x の DoS には、再帰ルックアップとコンテキストルックアップパターンが含まれている

3.6. ホストシステムおよび依存関係の問題

場合によっては、Data Grid のデプロイメントで、ホストシステムまたは外部の依存関係が原因でエラーが発生することがあります。このセクションでは、このような既知の問題の詳細と、トラブルシューティングおよび回避策の手順について説明します。

Red Hat Enterprise Linux 7 の TLS

RHEL 7 は、TLSv1.3 へのサポートをまだ提供していない OpenSSL ライブラリーのバージョンを提供します。しかし、Data Grid Server 8.2 はデフォルトで TLSv1.3 および TLSv1.2 を有効にします。これにより、暗号化された Hot Rod および REST エンドポイントのクライアント接続でエラーが発生します。

Data Grid Server は、以下のようなメッセージもログに記録します。

```
WARN [org.infinispan.HOTROD] ISPN004098: Closing connection due to transport error
org.infinispan.client.hotrod.exceptions.TransportException:: ISPN004077:
Closing channel due to error in unknown operation.
```

RHEL 7 に Data Grid Server をインストールする場合は、以下の JVM オプションで OpenSSL を無効にしてネイティブ Java SSL ライブラリーを使用する必要があります。

```
-Dorg.infinispan.openssl=false
```

関連情報

- [Securing Applications with TLS in RHEL](#)

第4章 テクノロジープレビュー

Data Grid リリースは、テクノロジープレビュー機能を提供します。これらの機能については Red Hat のサポートの詳細をご覧ください。

4.1. テクノロジープレビュー機能

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、テクノロジープレビュー機能の実稼働環境での使用を推奨していません。これらの機能により、近日発表予定の製品機能をリリースに先駆けてご提供でき、お客様は開発プロセス時に機能をテストして、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。