



Red Hat Data Grid 8.1

Red Hat Data Grid 8.1 Release Notes

Get release information for Data Grid 8.1

Red Hat Data Grid 8.1 Red Hat Data Grid 8.1 Release Notes

Get release information for Data Grid 8.1

Legal Notice

Copyright © 2021 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.

Abstract

Find out about features and enhancements in Data Grid 8.1, learn about current known issues and resolved issues, review the configurations that Red Hat supports, and more.

Table of Contents

CHAPTER 1. RED HAT DATA GRID	3
1.1. DATA GRID DOCUMENTATION	3
1.2. DATA GRID DOWNLOADS	3
1.3. MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 2. DATA GRID RELEASE INFORMATION	4
2.1. WHAT'S NEW IN DATA GRID 8.1.1	4
2.1.1. Data Grid Server	4
Customizable Server Filesystem	4
Set System Properties from File	4
Run as System Service	4
Port Offset Takes Effect for Cluster Transport	4
2.1.2. Data Grid Operator	5
Configuring Anti-Affinity for Data Grid Nodes	5
Disable Encryption	5
Install into Multiple Namespaces	5
2.1.3. Data Grid CLI Configuration	5
2.1.4. Improved Non-Blocking Thread Pool Usage	5
2.1.5. ProtoStream API Upgraded	5
2.1.6. Additions and Amendments to Documentation	6
2.2. WHAT'S NEW IN DATA GRID 8.1.0	6
2.2.1. Cross-Site Replication	6
2.2.2. Data Grid Server	7
2.2.3. Data Grid Operator	8
2.2.4. Data Grid Command Line Interface (CLI)	9
2.2.5. Data Grid Console	10
2.2.6. REST API	10
2.2.7. Hot Rod Java Clients	10
2.2.8. Data Grid Queries	11
2.2.9. Simplified Configuration	11
2.2.10. Non-Blocking I/O	11
2.2.11. Red Hat JBoss EAP Modules	12
CHAPTER 3. DATA GRID ON OPENSIFT	13
3.1. DATA GRID OPERATOR VERSION DETAILS	13
3.2. DATA GRID 8.1 IMAGES	14
Custom Data Grid Deployments	14
3.3. DATA GRID LIBRARY MODE ON OPENSIFT	14
CHAPTER 4. KNOWN AND FIXED ISSUES	16
4.1. KNOWN ISSUES FOR DATA GRID	16
4.1.1. Open Issues for Data Grid 8.1	17
Data Grid 8.1.1	17
Data Grid 8.1	17
4.2. FIXED IN DATA GRID 8.1.1	18
4.3. FIXED IN DATA GRID 8.1.0	19

CHAPTER 1. RED HAT DATA GRID

Data Grid is a high-performance, distributed in-memory data store.

Schemaless data structure

Flexibility to store different objects as key-value pairs.

Grid-based data storage

Designed to distribute and replicate data across clusters.

Elastic scaling

Dynamically adjust the number of nodes to meet demand without service disruption.

Data interoperability

Store, retrieve, and query data in the grid from different endpoints.

1.1. DATA GRID DOCUMENTATION

Documentation for Data Grid is available on the Red Hat customer portal.

- [Data Grid 8.1 Documentation](#)
- [Data Grid 8.1 Component Details](#)
- [Supported Configurations for Data Grid 8.1](#)
- [Data Grid 8 Feature Support](#)
- [Data Grid Deprecated Features and Functionality](#)

1.2. DATA GRID DOWNLOADS

Access the [Data Grid Software Downloads](#) on the Red Hat customer portal.



NOTE

You must have a Red Hat account to access and download Data Grid software.

1.3. MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 2. DATA GRID RELEASE INFORMATION

Learn about new features and get the latest Data Grid release information.

2.1. WHAT'S NEW IN DATA GRID 8.1.1

Data Grid 8.1.1 gives you usability and performance improvements and enhances security. Review all the enhancements and updates that are available in 8.1.1.

2.1.1. Data Grid Server

This release includes several enhancements to Data Grid Server.

Customizable Server Filesystem

Data Grid Server now includes system properties that override the default location for folders under the server root directory.

Folder	Description	System property override
/server/conf	Contains server configuration files.	infinispan.server.config.path
/server/data	Contains data files organized by container name.	infinispan.server.data.path
/server/lib	Contains server extension files. This directory is scanned recursively and used as a classpath.	infinispan.server.lib.path Separate multiple paths with the following delimiters: : on Unix / Linux ; on Windows
/server/log	Contains server log files.	infinispan.server.log.path

- [Data Grid Server Filesystem](#)
- [Server root directory](#)

Set System Properties from File

You can now use set system properties for Data Grid Server from file with the **-P** and **--properties** options, for example:

```
$ ./bin/server.sh -P path/to/system.properties
```

Run as System Service

Data Grid Server provides system **init** files so you can run server instances as services on Red Hat Enterprise Linux.

See [Run JBoss Data Grid 7 as a RHEL 7 Service](#) (Red Hat Knowledgebase) for more information.

Port Offset Takes Effect for Cluster Transport

When you specify a port offset for Data Grid Server, that port offset now applies to **jgroups.bind.port** and **jgroups.mcast.port**.

- [Specifying Port Offsets](#)

2.1.2. Data Grid Operator

This release includes several enhancements to Data Grid Operator.

Configuring Anti-Affinity for Data Grid Nodes

Anti-affinity settings protect workloads from single points of failure.

Data Grid Operator provides a **spec.affinity** field in your **Infinispan** CR that configures anti-affinity capabilities.

- [Guaranteeing Availability with Anti-Affinity](#)

Disable Encryption

You can now disable encryption for client connections.

```
security:
  endpointEncryption:
    type: None
```

- [Disabling Encryption](#)

Install into Multiple Namespaces

As of this release, you can install Data Grid Operator into specific namespaces or multiple namespaces.

- [Installing Data Grid Operator](#)

2.1.3. Data Grid CLI Configuration

Configure the Command Line Interface (CLI) with command aliases and custom startup properties. You can also define the location on your filesystem where the CLI stores configuration.

- [Configuring the Data Grid CLI](#)

2.1.4. Improved Non-Blocking Thread Pool Usage

In previous releases, Data Grid used a Java **ThreadPoolExecutor** class to manage the blocking thread pool. As of this release, Data Grid replaces **ThreadPoolExecutor** with an **EnhancedQueueExecutor** class that allocates new threads only if there are no idle threads. This change improves the efficiency of the non-blocking thread pool usage by ensuring that threads are allocated only when necessary.

2.1.5. ProtoStream API Upgraded

Data Grid 8.1.1 upgrades the ProtoStream API 4.3.4.



IMPORTANT

In previous versions, the ProtoStream API did not correctly nest message types with the result that the messages were generated as top-level only. For this reason, if you have Protobuf messages in a persistent cache store and upgrade to Data Grid 8.1.1, then you should modify Java classes so that Protobuf annotations are at top-level. This ensures that the nesting in your persisted messages matches the nesting in your Java classes, otherwise data incompatibility issues can occur.

2.1.6. Additions and Amendments to Documentation

Data Grid documentation is updated to include the following content:

- [Data Grid Library Mode on OpenShift](#) is updated with additional information to improve clarity.
- [Getting Started with Data Grid Server](#) contains updated information and is restructured for better findability.
- Documentation for configuring cluster transport with JGroups protocols is now removed from the *Configuration Guide* and included in [Server Guide: Setting Up Data Grid Clusters](#) and [Library Mode: Setting Up Data Grid Clusters](#). These changes are intended to provide cluster transport configuration specific to embedded and remote cache usage.
- [Using JGroups System Properties](#) is updated for configuring cluster transport.
- [Encrypting Cluster Transport](#) documentation is now available in the *Security Guide* as well as in sections for configuring cluster transport.
- [Clustered Locks](#) documentation is updated to remove incorrect details and improve clarity.
- [Configuring Data Grid to Marshall Java Objects](#) is rewritten to provide clear procedures for generating Serialization Context Initializers and details about default supported types.
- [Creating a Prometheus Service Monitor](#) were updated to add missing attributes.
- [Installing Data Grid Operator from the Command Line](#) was amended so that the step to verify installation does not require users to switch projects.

2.2. WHAT'S NEW IN DATA GRID 8.1.0

Data Grid 8.1.0 improves usability, increases performance, and enhances security. Find out what's new in 8.1.0.

2.2.1. Cross-Site Replication

Full Support on Red Hat OpenShift

Data Grid cross-site replication is now fully supported with Data Grid Operator. Create global Data Grid clusters on OpenShift that back up across your data centers.

See [Configuring Cross-Site Replication on OpenShift](#).

Conflict Resolution for Asynchronous Backups

Data Grid 8.1 significantly improves asynchronous backups when using cross-site replication capabilities. This release introduces detection for conflicting entries and a resolution mechanism to guarantee data consistency when you have multiple active sites replicating between each other.

Find out complete details and learn how Data Grid detects and resolves conflicts in the documentation.

See [Concurrent Writes and Conflicting Entries](#).

Multiple Site Masters

Data Grid uses JGroups RELAY2 protocol to facilitate inter-cluster communication for cross-site replication. Each cluster has a site master that handle traffic between sites.



NOTE

Data Grid Operator controls which nodes become site master when using cross-site replication on OpenShift. You cannot modify the RELAY2 configuration with Data Grid Operator.

As of 8.1, you can set multiple site masters in your RELAY2 configuration to balance inter-cluster traffic. Use the **max_site_masters** attribute as in the following example:

```
<relay.RELAY2 site="NYC" 1
    max_site_masters="100"/> 2
```

- 1** Defines the name of the local site.
- 2** Specifies the number of site masters. The default value is **1**. You can increase the value to any number greater than one. If you set a value that is equal to or greater than the number of nodes in your Data Grid cluster then all nodes act as site masters.

See [Setting Up Cross-Site Replication](#).

2.2.2. Data Grid Server

Data Grid Server 8.1 offers enhanced security and new configuration options.

Automatic Authentication

Data Grid Server uses a default property realm that automatically enforces user authentication on remote endpoints.

See [Adding Data Grid Credentials](#) to create usernames and passwords for accessing the Data Grid Console, Command Line Interface (CLI), REST API, or Hot Rod endpoint.

See [Securing Access to Data Grid Servers](#) to learn about configuring different authentication mechanisms.



NOTE

If you installed Data Grid Server 8.0, upgrade to 8.1 and use the configuration schema with default security.

Shared Data Sources for JDBC Cache Stores

Data Grid Server lets you create shared data sources when using JDBC cache stores to persist cache entries to databases.

See [Configuring Data Grid Server Data Sources](#).

Global Roles for Cache Authorization

If you configure cache authorization with role-based access, roles that you declare in the global configuration apply unless you explicitly declare roles in your caches.

```

</infinispan>
  <cache-container name="default">
    <security>
      <authorization> 1
        <identity-role-mapper/>
        <role name="AdminRole" permissions="ALL"/>
        <role name="ReaderRole" permissions="READ"/>
        <role name="WriterRole" permissions="WRITE"/>
        <role name="SupervisorRole" permissions="READ WRITE EXEC BULK_READ"/>
      </authorization>
    </security>
    <distributed-cache name="secured">
      <security>
        <authorization/> 2
      </security>
    </distributed-cache>
  </cache-container>
  ...
</infinispan>

```

- 1** Defines user roles that map to permissions.
- 2** Implicitly uses all authorization roles in the global configuration.

See [Configuring Data Grid Authorization](#).

2.2.3. Data Grid Operator

Data Grid Operator 8.1 provides new features and improvements for running Data Grid on Red Hat OpenShift.

Cross-Site Replication

Data Grid Operator improves setup and management of cross-site replication capabilities for Data Grid clusters running on OpenShift.

See [Cross-Site Replication with Data Grid Operator](#).

Expose Services via Routes

Data Grid Operator updates the **spec.expose** resource so you can create OpenShift Routes with passthrough encryption to make Data Grid clusters available on the network.

See [Exposing Data Grid Through Routes](#).

Automatic Scaling

Data Grid Operator can automatically scale the default cache on Cache service nodes up or down based on memory usage.

See [Configuring Automatic Scaling](#).

Automatic Encryption with the OpenShift Service CA

By default, if the Red Hat OpenShift service CA is available, Data Grid Operator generates TLS certificates, signed by the Red Hat OpenShift service CA, to encrypt client connections.

**NOTE**

You must use encryption if the OpenShift service CA is present.

If you are upgrading from 8.0, you should retrieve the generated **tls.crt** certificate and add it to a client trust store.

See [Securing Data Grid Connections](#).

Cache Custom Resource

You can now create caches with Data Grid service nodes with the **Cache** custom resource.

**IMPORTANT**

Creating caches with Data Grid Operator is available as a technology preview.

Technology Preview features or capabilities are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

See [Creating Data Grid Caches](#).

Number of Owners

Cache service nodes now use a default value of **2** for the number of owners, which replicates each entry across the cluster. You can modify the number of owners with the **spec.service.replicationFactor** resource in your **Infinispan** CR.

See [Configuring the Number of Owners](#).

2.2.4. Data Grid Command Line Interface (CLI)

Data Grid 8.1 CLI provides additional capabilities for server configuration, user management, and troubleshooting.

User Management

As of 8.1, the CLI provides a **user** command that replaces the **user-tool** script included in Data Grid Server distributions.

The **user** command lets you manage user credentials and groups for property-based security realms. For example, the following commands adds credentials that have "admin" privileges:

```
[disconnected]> user create myusername --password=qwer1234! --groups=admin
```

See [Adding Data Grid Credentials](#).

Logging Configuration

You can now modify server logging configuration at runtime. For example, to enable **DEBUG** log messages for the **org.infinispan** category:

```
[hostname@cluster//containers/default]> logging set --level=DEBUG org.infinispan
```

See [Changing Data Grid Server Logging Configuration at Runtime](#) .

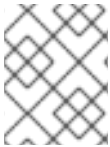
Diagnostic Reports

Data Grid Server provides an aggregate log that includes information, such as thread dumps, memory configuration, and open sockets, that you can use to monitor and troubleshoot deployments.

Use the **report** command with the CLI as in the following example:

```
[hostname@cluster//containers/default]> server report
Downloaded report 'infinispan-hostname-timestamp-report.tar.gz'
```

See [Getting Diagnostic Reports for Data Grid Servers](#) .



NOTE

Diagnostics reports for Data Grid Server are currently available on Linux/ Unix systems only.

2.2.5. Data Grid Console

Data Grid 8.1 Console provides additional capabilities such as:

- Creating and deleting cache entries.
- Cross site replication management.
- Schema management.
- Searching the data container with Ickle queries.
- Integration with Red Hat SSO so that accessing Data Grid Console obtains credentials through the realm login page.

For information on accessing the Data Grid Console and creating caches, see [Creating Caches with the Data Grid Console](#).

2.2.6. REST API

Calling the Data Grid REST API with the **?action** URL parameter is now supported with the **POST** method. Calls return status **200** if responses contain content or status **204** if not.

Support for the **?action** URL parameter with the **GET** method is planned for removal.

2.2.7. Hot Rod Java Clients

Configuration via URI

Hot Rod Java Client configuration now supports URI in the following format:

```
hotrod[s]://[user[:password]@]host[:port][,host2[:port]][?property=value[&property2=value2]]
```

This provides a more compact way for configuring your Hot Rod Java client connections. See [Hot Rod Client Configuration](#).

Per-Cache Configuration with Automatic Cache Creation

When Hot Rod Java clients attempt to access caches that do not exist, they return null for `getCache("<cache_name>")` invocations. You can change this default behavior so that clients automatically create caches on first access using caches configuration templates or Data Grid cache definitions in XML format.

See [Creating Caches on First Access](#).

2.2.8. Data Grid Queries

Data Grid indexing and querying capabilities have undergone changes that decrease configuration overhead and offer better performance when searching the data store.

Major changes include:

- Deprecation of the Data Grid Lucene Directory.
- Deprecation of the `org.infinispan.query.SearchManager` interface.

Data Grid no longer supports native objects for Apache Lucene and Hibernate Search. Use the `org.infinispan.query.Search` class as your entry point for string-based searches with the Ickle query language.



IMPORTANT

You should plan to migrate any usage of deprecated search functionality, including Lucene-based queries, and start using the `org.infinispan.query.Search` class as an entry point for performing searches with the Ickle query language.

See the following:

- [Data Grid Search Documentation](#)
- [Search Javadoc](#)

2.2.9. Simplified Configuration

Data Grid 8.1 deprecates several configuration attributes to streamline user options and improve consistency between declarative and programmatic settings.

memory configuration is the most notable change in Data Grid 8.1. It lets you configure the data container to:

- Store entries in the JVM heap or in off-heap memory.
- Control the size of the data container. You can set a maximum number of entries that caches can hold or set a maximum amount of memory that caches can use.

As in previous releases, the **memory** configuration can also configure Data Grid to store entries in binary format. However, as of 8.1, binary storage configuration is deprecated. Instead you should define the encoding for caches. If you want to use binary storage, specify the `MediaType` for any binary format in your cache definition.

2.2.10. Non-Blocking I/O

Data Grid consolidates thread pools into two:

- Non-blocking thread pool
- Blocking thread pool

As of this release, nearly all Data Grid internal components use the non-blocking thread pool, which increases throughput, reduces latency, and improves overall responsiveness for heavy I/O workloads.

Any blocking I/O paths, now use the blocking thread pool to avoid holding up non-blocking calls.

NonBlockingStore SPI

Data Grid 8.1 introduces the **NonBlockingStore** interface for implementing persistent cache stores. As the name implies **NonBlockingStore** SPI exposes methods that must never block the invoking thread.

To handle blocking operations, Data Grid provides a **BlockingManager** utility class.

See the following:

- [Setting Up Persistent Storage](#)
- [NonBlockingStore Javadoc](#)

2.2.11. Red Hat JBoss EAP Modules

Data Grid Modules for Red Hat JBoss EAP provide a temporary solution until planned functionality for Red Hat JBoss EAP to directly manage the **infinispan** subsystem is available.

See [Data Grid Modules for Red Hat JBoss EAP](#) .

CHAPTER 3. DATA GRID ON OPENSHIFT

3.1. DATA GRID OPERATOR VERSION DETAILS

The following table provides version information for Data Grid Operator.



NOTE

Data Grid Operator versions do not directly correspond to Data Grid versions because the release schedule is more frequent.

Data Grid Operator version	Data Grid version	Features
8.1.6	8.1.1	<ul style="list-style-type: none"> * Bug fixes. * Configurable StorageClass objects for persistent volume claims. * Documentation improvements for creating routes, Hot Rod client configuration, and cache creation. * Fixes security vulnerabilities.
8.1.5	8.1.1	<ul style="list-style-type: none"> * Support for custom labels. * Metering labels for Red Hat subscriptions. * Fixes security vulnerabilities.
8.1.4	8.1.1	<ul style="list-style-type: none"> * Anti-affinity settings. * Ability to disable encryption. * Multiple namespace installation. * Data Grid 8.1.1
8.1.3	8.1.0	Fixes security vulnerabilities.
8.1.2	8.1.0	Fixes security vulnerabilities.
8.1.1	8.1.0	Fixes security vulnerabilities.

Data Grid Operator version	Data Grid version	Features
8.1.0	8.1.0	<ul style="list-style-type: none"> * Cross-site replication. * Automatic scaling. * Ability to expose services via routes. * Automatic encryption with the OpenShift service CA. * Cache CR. * Configurable number of owners for Cache service nodes. * Data Grid 8.1.0

3.2. DATA GRID 8.1 IMAGES

Data Grid 8.1 includes two container images, the Data Grid Operator image and Data Grid Server image.

Data Grid images are hosted on the Red Hat Container Registry, where you can find health indexes for the images along with information about each tagged version.



NOTE

Red Hat supports Data Grid 8.1 on OpenShift only through Data Grid Operator subscriptions.

Custom Data Grid Deployments

Red Hat does not support customization of any 8.1 images from the Red Hat Container Registry through the Source-to-Image (S2I) process or **ConfigMap** API.

As a result it is not possible to use custom:

- Discovery protocols
- Encryption mechanisms (SYM_ENCRYPT or ASYM_ENCRYPT)
- Persistent datasources

Likewise Data Grid Operator does not allow you to deploy custom code (**JAR** files or other artefacts) to Data Grid.

Additional resources

- [Data Grid Container Images](#)

3.3. DATA GRID LIBRARY MODE ON OPENSIFT

Embedding Data Grid in custom applications, also referred to as Library Mode, is intended for specific uses only when running on OpenShift:

- Using local or distributed caching in custom Java applications to retain full control of the cache lifecycle. Additionally, when using features that are available only with embedded Data Grid such as distributed streams.
- Reducing network latency to improve the speed of cache operations.

The Hot Rod protocol provides near-cache capabilities that achieve equivalent performance to a standard client-server architecture.

Requirements

Embedding Data Grid in applications running on OpenShift requires you to use a discovery mechanism so Data Grid nodes can form clusters to replicate and distribute data.

Red Hat supports only DNS_PING as the cluster discovery mechanism.

DNS_PING exposes a port named **ping** that Data Grid nodes use to perform discovery and join clusters. TCP is the only supported protocol for the **ping** port, as in the following example for a pod on OpenShift:

```
spec:
  ...
  ports:
  - name: ping
    port: 8888
    protocol: TCP
    targetPort: 8888
```

Limitations

Embedding Data Grid in applications running on OpenShift also has some specific limitations:

- Persistent cache stores are not currently supported.
- UDP is not supported with embedded Data Grid.

Custom caching services

Red Hat highly discourages embedding Data Grid to build custom caching servers to handle remote client requests. To benefit from regular, automatic updates with performance improvements and fix security issues, you should create Data Grid clusters with the Data Grid Operator instead.

Reference

- [DNS_PING](#)

CHAPTER 4. KNOWN AND FIXED ISSUES

Learn about known issues in Data Grid and find out which issues are fixed.

4.1. KNOWN ISSUES FOR DATA GRID

IllegalArgument Exception when performing rolling upgrades with transactional caches

Issue: [JDG-4315](#)

Description: It is not possible to perform a rolling upgrade with transactional caches. Upgrades fail and the following exception is written to logs:

```
Caused by: java.lang.IllegalArgumentException: Cannot create a transactional context without a
valid Transaction instance.
    at
org.infinispan.context.impl.TransactionInvocationContextFactory.createInvocationContext(Transact
onalInvocationContextFactory.java:63)
[...]
    at org.infinispan.cache.impl.DecoratedCache.putIfAbsent(DecoratedCache.java:688)
    at
org.infinispan.cache.impl.AbstractDelegatingAdvancedCache.putIfAbsent(AbstractDelegatingAdvanc
edCache.java:328)
    at org.infinispan.cache.impl.EncoderCache.putIfAbsent(EncoderCache.java:450)
    at
org.infinispan.persistence.remote.upgrade.MigrationTask.lambda$migrateEntriesWithMetadata$0(
MigrationTask.java:128)
    at
java.base/java.util.concurrent.CompletableFuture$AsyncSupply.run(CompletableFuture.java:1771)
```

Workaround: There is no workaround for this issue.

IOException occurs with Data Grid CLI on OpenShift

Issue: [JDG-4292](#)

Description: Running the Command Line Interface (CLI) from Data Grid pods running on OpenShift results in an **java.io.IOException: Permission denied** message.

For example, if you connect to the pod using **oc rsh <pod-name>** and then invoke the CLI from the **/opt/infinispan/bin** directory, when you quit the CLI the exception is thrown.

This exception is related to the Aesh component and can be safely ignored.

Workaround: Download Data Grid Server and use the CLI on your local host system.

```
$ bin/cli.sh -c https://$SERVICE_HOSTNAME:11222/ --trustall
```

Data Grid on OpenShift continually restart after OOM exceptions

Issue: [JDG-3991](#)

Description: If out of memory exceptions cause Data Grid Server to terminate on OpenShift, the nodes cannot restart. The following exception is written to the pod log file:

```
FATAL (main) [org.infinispan.SERVER] ISPN080028: Red Hat Data Grid Server failed to start
java.util.concurrent.ExecutionException:
```

org.infinispan.manager.EmbeddedCacheManagerStartupException:
 org.infinispan.commons.CacheException: Initial state transfer timed out for cache
 org.infinispan.LOCKS on <pod-name-id>

Workaround: There is no workaround for this issue.

Cannot authenticate with Data Grid Server from HotRod Store or NodeJS clients

Issue: [JDG-3868](#)

Description: If you use the HotRod store on Red Hat JBoss EAP to externalize session data to Data Grid, or if you use the Hot Rod NodeJS client, you cannot connect to Data Grid servers that require client authentication, which is the default configuration.

Workaround: Use the **remote-store** implementation with Red Hat JBoss EAP 7.3 in combination with server authentication. To use the HotRod store on Red Hat JBoss EAP, or to use Hot Rod NodeJS clients, you must disable Data Grid Server authentication.

Inconsistent query behavior when indexed caches contain non-indexed Protobuf entities

Issue: [JDG-3972](#)

Description: If a cache is indexed but contains a Protobuf entity that is not indexed, search operations on that cache return inconsistent results.

Workaround: There is no workaround for this issue.

Data Grid Conflict Resolution Performance

Issue: [JDG-3636](#)

Description: In some test cases, Data Grid partition handling functionality took longer than expected to perform conflict resolution.

Workaround: There is no workaround for this issue.

Data Grid Does Not Passivate JWS Sessions Correctly

Issue: [JDG-2796](#)

Description: When externalizing sessions from JBoss Web Server (JWS), sessions are not passivated correctly if using the **FINE** persistence strategy.

Workaround: There is no workaround for this issue.

4.1.1. Open Issues for Data Grid 8.1

You can find the complete list of issues for Data Grid with the following Jira query:

Data Grid 8.1.1

[List of All Open Issues for Data Grid 8.1](#)

Query JQL

```
project = JDG AND affectedVersion in ("RHDG 8.1.1 CR1", "RHDG 8.1.1 CR2", "RHDG 8.1.1 GA")
AND status not in (Closed, Verified, "Ready for QA", "QA In Progress", Resolved) AND type = Bug
```

Data Grid 8.1

[List of All Open Issues for Data Grid 8.1](#)

Query JQL

```
project = JDG AND affectedVersion in ("RHDG 8.1 CD", "RHDG 8.1 CR1", "RHDG 8.1 CR2", "RHDG 8.1 GA") AND status not in (Closed, Verified, "Ready for QA", "QA In Progress", Resolved) AND type = Bug
```

4.2. FIXED IN DATA GRID 8.1.1

[List of All Issues Resolved in Data Grid 8.1.1](#)

Features and enhancements, documentation improvements, bug fixes.

Query JQL

```
project = JDG AND fixVersion in ("RHDG 8.1.1 GA", "RHDG 8.1.1 CR1", "RHDG 8.1.1 CR2")
```

Notable Fixes

This release includes the following notable fixes:

- [JDG-4274](#) Rolling upgrade operations do not work with caches that store POJOs "application/x-java-objects".
- [JDG-4262](#) Exceptions occur when iterating over embedded caches when partition handling is configured.
- [JDG-4270](#) The metrics endpoint calls size when gathering statistics which degrades performance.
- [JDG-4212](#) Possible deadlocks occur if an entry is replaced at the same time expiration takes effect.
- [JDG-4202](#) Console shows a blank page when cache authorization is configured.
- [JDG-3367](#) Hot Rod C++ client cannot be installed on Red Hat Enterprise Linux 8.
- [JDG-4043](#) Potential race condition exists in EncoderCache.
- [JDG-4062](#) Hot Rod C++ client behaves unexpectedly when server not running.
- [JDG-4116](#) Data Grid Operator upgrade process fails when upgrading from 8.0.
- [JDG-4162](#) Implicit authentication configuration overwrites custom authentication configuration.
- [JDG-4177](#) Server garbage collection logging configuration not valid with Java 11.
- [JDG-4200](#) REST endpoint provides incomplete chunked static resources.
- [JDG-3989](#) Performance degradation occurs if backup locations become unavailable.
- [JDG-4016](#) Data Grid Operator overwrites creation timestamps.
- [JDG-3956](#) Null pointer exception occurs when accessing data during state transfer operations.

- [JDG-3971](#) Excessive messages written to log files when writing to caches with offline backups.
- [JDG-3494](#) Cannot Access Caches If Names Contain the "/" Character.
- [JDG-3504](#) Remote Cache Stores Cannot Use Preload with EAP Session Externalization.

4.3. FIXED IN DATA GRID 8.1.0

List of All Issues Resolved in Data Grid 8.1

Features and enhancements, documentation improvements, bug fixes.

Query JQL

```
project = JDG AND fixVersion in ("RHDG 8.1 CD", "RHDG 8.1 CR1", "RHDG 8.1 CR2", "RHDG 8.1 GA")
```

Notable Fixes

This release includes the following notable fixes:

- [JDG-3826](#) Global persistent locations do not retrieve server data path property.
- [JDG-3591](#) Data Grid Operator does not encrypt endpoints by default.
- [JDG-3624](#) Data Grid Server allows REST and Hot Rod operations from unauthenticated users
- [JDG-3799](#) Cache configuration exception caused by PartitionStrategy must be ALLOW_READ_WRITES when numOwners is one.
- [JDG-3738](#) Uneven segment distributions can happen after the rebalancing with SyncConsistentHashFactory.
- [JDG-3675](#) Logging through stdout includes colour codes.
- [JDG-3586](#) Console creates a local cache if no configuration is provided.