Red Hat Data Grid 8.1

Red Hat Data Grid 8.1 Release Notes

Get release information for Data Grid 8.1

Last Updated: 2023-11-23

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

Get release information for Data Grid 8.1

Legal Notice

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

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.

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

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.

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 1. UPGRADE TO THE LATEST DATA GRID VERSION

Red Hat recommends you upgrade any deployments from 8.1.x to the latest Data Grid 8 version as soon as possible. The Data Grid team regularly patch security vulnerabilities and actively fix issues on the latest version of the software.

1.1. APACHE LOG4J SECURITY VULNERABILITIES

Data Grid includes Apache Log4j components in:

- Data Grid Server distribution

- Data Grid Server container image for Red Hat OpenShift

Data Grid 8.2.2 fixes CVE-2021-44228 which is a critical security vulnerability. Data Grid 8.2.3 fixes several Log4j CVEs of moderate severity. For more information, see the Data Grid 8.2 release notes or the Data Grid Operator 8.2 release notes. You can find the latest Data Grid documentation at Red Hat Data Grid Product Documentation.

> **NOTE**
>
> Data Grid Operator version 8.2.7 corresponds to Data Grid 8.2.2.

If you cannot upgrade your Data Grid deployment, you must follow the steps to mitigate the critical vulnerability as outlined in the RHSB-2021-009 Log4Shell – Remote Code Execution security bulletin.

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> ❶
        <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/> ❷
      </security>
    </distributed-cache>
  </cache-container>
  ...
</infinispan>
```

❶   Defines user roles that map to permissions.

❷   Implicitly uses all authorization roles in the global configuration.

See Configuring Data Grid Authorization.

2.2.3. 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.

2.2.4. 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.

2.2.5. 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.

2.2.6. 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 .

2.2.7. 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

2.2.8. 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.

2.2.9. 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

2.2.10. 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 .

2.3. SUPPORTED JAVA VERSIONS IN DATA GRID 8.1

Red Hat supports different Java versions, depending on how you install Data Grid.

Embedded caches
Red Hat supports Java 8 and Java 11 when using Data Grid for embedded caches in custom applications.

Remote caches
Red Hat supports Java 11 only for Data Grid Server installations. For Hot Rod Java clients, Red Hat supports Java 8 and Java 11.

Java 8 deprecation
Support for Java 8 is deprecated in the next release, Data Grid 8.2, and currently planned for removal in Data Grid 8.4.

Users with embedded caches in custom applications should plan to upgrade to Java 11 or to Java 17 when support becomes available.

Hot Rod Java clients running in applications that require Java 8 can continue using older versions of client libraries. Red Hat supports using older Hot Rod Java client versions in combination with the latest Data Grid Server version.

Additional resources

- Supported Configurations for Data Grid 8.1

- Data Grid Deprecated Features and Functionality

CHAPTER 3. KNOWN AND FIXED ISSUES

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

3.1. KNOWN ISSUES FOR DATA GRID

IllegalArgumentException 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.TransactionalInvocationContextFactory.createInvocationContext(TransactionalInvocationContextFactory.java:63)
[...]
    at org.infinispan.cache.impl.DecoratedCache.putIfAbsent(DecoratedCache.java:688)
    at org.infinispan.cache.impl.AbstractDelegatingAdvancedCache.putIfAbsent(AbstractDelegatingAdvancedCache.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.

3.2. FIXED IN DATA GRID 8.1.1

Data Grid 8.1.1 includes the following 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.

3.3. FIXED IN DATA GRID 8.1.0

Data Grid 8.1.0 includes the following 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.