



Red Hat Data Grid 8.3

Red Hat Data Grid 8.3 Release Notes

Get release information for Data Grid 8.3

Red Hat Data Grid 8.3 Red Hat Data Grid 8.3 Release Notes

Get release information for Data Grid 8.3

Legal Notice

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.

Abstract

Find out about features and enhancements in Data Grid 8.3 as well as known and resolved issues.

Table of Contents

RED HAT DATA GRID	4
DATA GRID DOCUMENTATION	5
DATA GRID DOWNLOADS	6
MAKING OPEN SOURCE MORE INCLUSIVE	7
CHAPTER 1. DATA GRID RELEASE INFORMATION	8
1.1. WHAT'S NEW IN DATA GRID 8.3.0	8
1.1.1. Java 17 support	8
1.1.2. Data Grid Helm chart	8
1.1.3. Data Grid configuration	8
Polyglot configuration	8
Mutable configuration attributes for dynamic caches	8
Cache size statistics	8
1.1.4. Persistent storage	9
Default file-based cache store implementation	9
SQL cache store	10
JDBC connection factories	10
1.1.5. Data Grid Server	10
Distributed security realms	10
Multiple endpoint configuration	11
Security enhancements	11
Configuration overlays	12
Listening on all addresses	12
Performance and usability improvements	13
1.1.6. Data Grid Console	13
1.1.7. Data Grid Command Line Interface (CLI)	13
Modifying configuration attributes at runtime	13
Controlling cache availability	13
Enabling and disabling rebalancing operations	14
Hot Rod cluster migration	14
Retrieving cross-site replication relay nodes	14
Native CLI	14
1.1.8. REST API	14
Modifying configuration attributes at runtime	14
Controlling cache availability	15
Enabling and disabling rebalancing operations	15
Hot Rod migration	15
Retrieving cross-site replication relay nodes	15
Converting cache configuration between XML, JSON, and YAML	16
Cache entry and cache configuration listeners with server-sent events	16
1.1.9. Query API	16
Delete by query	16
Query execution methods	16
1.1.10. Cross-site replication	16
Improved observability and metrics	16
Naming global clusters	17
1.1.11. Rolling upgrades and Hot Rod migration	17
1.2. WHAT'S NEW IN DATA GRID 8.3.1	17
Compliance with the Federal Information Processing Standards (FIPS) 140-2	17

Improved metrics	18
Approximate cache size statistics	18
Remote site status metrics for cross-site connection	18
JGroups attributes exposed through the metrics endpoint	18
Performance enhancements to cross-site replication	18
1.3. SUPPORTED JAVA VERSIONS IN DATA GRID 8.3	18
Embedded caches	18
Remote caches	19
Java 8 deprecation	19
CHAPTER 2. KNOWN AND FIXED ISSUES	20
2.1. KNOWN ISSUES FOR DATA GRID	20
2.2. FIXED IN DATA GRID 8.3.0	21
2.3. FIXED IN DATA GRID 8.3.1	21
2.4. HOST SYSTEM AND DEPENDENCY ISSUES	22
TLS on Red Hat Enterprise Linux 7	22

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 8.3 Documentation](#)
- [Data Grid 8.3 Component Details](#)
- [Supported Configurations for Data Grid 8.3](#)
- [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. DATA GRID RELEASE INFORMATION

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

1.1. WHAT'S NEW IN DATA GRID 8.3.0

Data Grid 8.3 improves usability, increases performance, and enhances security. Find out what's new.

1.1.1. Java 17 support

Data Grid 8.3 offers full support for Java 17 for embedded and remote caches.

1.1.2. Data Grid Helm chart

Data Grid 8.3 introduces a Helm chart for deploying Data Grid clusters on Red Hat OpenShift. The Data Grid Helm chart provides an alternative for scenarios where it is not possible to deploy clusters that the Data Grid Operator manages, or where you require manual configuration, deployment, and management of Data Grid clusters.

The Data Grid Helm chart includes **deploy.infinispan** values that let you apply custom configuration to Data Grid Server nodes with the Helm client or through the OpenShift Web Console.

1.1.3. Data Grid configuration

Data Grid provides declarative and programmatic configuration attributes for customizing caches and underlying mechanisms such as security and cluster transport.

Polyglot configuration

Data Grid 8.3 lets you create declarative configuration in JSON and YAML format in addition to XML. You can re-use the same Data Grid configuration files for any deployment platform: Red Hat Enterprise Linux (RHEL) or Red Hat OpenShift via Data Grid Operator or the Data Grid Helm chart.

Mutable configuration attributes for dynamic caches

Data Grid 8.3 lets you modify certain configuration attributes for caches that you create dynamically at runtime. This means you no longer need to delete and then recreate caches to change configuration attributes such as **lifespan** for expiration or **max-count** for eviction.

Changes to cache configuration must be compatible with the existing configuration. For example you cannot modify cache configuration to change the **replicated-cache** element to **distributed-cache** or use the **max-count** attribute instead of the **max-size** attribute.

Changing mutable configuration attributes

The Data Grid Command Line Interface (CLI) and REST API provide mechanisms for:

- Retrieving the list of mutable configuration attributes.
- Changing mutable configuration attributes at runtime.

For usage details, see the [CLI](#) or [REST](#) sections.

Cache size statistics

As of Data Grid 8.3, the **currentNumberOfEntries** statistic returns **-1** by default to improve performance.

Data Grid also now provides an **accurate-size** attribute for metrics configuration if you want to restore accurate values for this statistic. However this is a resource-intensive calculation and is recommended for development environments only. Set the value to **true** as follows:

Declarative

```
<cache-container>
  <metrics accurate-size="true"/>
</cache-container>
```

Programmatic

```
GlobalConfigurationBuilder b = new GlobalConfigurationBuilder();
b.metrics().accurateSize(true);
```



NOTE

Metrics that provide approximate cache size statistics are planned for a future version of Data Grid.

1.1.4. Persistent storage

Data Grid 8.3 provides new and improved cache store implementations for using persistent storage.

Default file-based cache store implementation

To improve performance by using less memory resources and avoiding fragmentation, **SoftIndexFileStore** is now the default implementation for file-based cache stores in Data Grid. As of 8.3, including **file-store** persistence in cache configuration creates a soft index file-based cache store instead of a single-file cache store.

Data Grid automatically migrates any **file-store** configuration to a **SoftIndexFileStore** when you upgrade to 8.3.

If you are migrating to Data Grid 8.3 from an earlier version and your cache configuration includes the **soft-index-file-store** element, you should convert it to the **file-store** element. For more information, see [Migrating to Data Grid 8](#).

Data Grid 8.2

```
<persistence>
  <soft-index-file-store xmlns="urn:infinispan:config:soft-index:12.1">
    <index path="testCache/index" />
    <data path="testCache/data" />
  </soft-index-file-store>
</persistence>
```

Data Grid 8.3

```
<persistence>
  <file-store>
    <index path="testCache/index" />
```

```
<data path="testCache/data" />
</file-store>
</persistence>
```

Additional resources

- [File-based cache stores](#)

SQL cache store

Data Grid 8.3 introduces the SQL cache store that simplifies access to data in persistent storage and pre-loads caches through JDBC connections.

You can create two types of SQL cache store:

Table

Data Grid loads entries from a single database table.

Query

Data Grid uses SQL queries to load entries as well as perform insert, update, and delete operations.

Additional resources

- [SQL cache stores](#)

JDBC connection factories

As of Data Grid 8.3 Data Grid provides **ConnectionFactory** implementations for JDBC connections in the **org.infinispan.persistence.jdbc.common.configuration** package. You can use these connection factories with SQL cache stores and JDBC string-based caches stores.

Additional resources

- [JDBC connection factories](#)

1.1.5. Data Grid Server

Data Grid Server runs in a dedicated Java Virtual Machine (JVM) and provides scalable and fault tolerant clusters for remote caches.

Distributed security realms

Data Grid Server now lets you combine multiple security realms into a single realm. When authenticating users, Data Grid Server uses each security realm in turn until it finds one that can perform the authentication.

For example, the following security realm includes an LDAP realm and a property realm and the **distributed-realm** element:

```
<security-realms>
  <security-realm name="my-distributed-realm">
    <ldap-realm>
      <!-- LDAP realm configuration. -->
    </ldap-realm>
    <properties-realm>
      <!-- Property realm configuration. -->
    </properties-realm>
```

```

    <distributed-realm/>
  </security-realm>
</security-realms>

```

Multiple endpoint configuration

You can configure Data Grid Server to bind multiple endpoints to different sockets and use different security realms.

The following snippet, as an example, contains two different endpoint configurations. One endpoint binds to a "public" socket, uses an "application" security realm, and disables administrative features. Another endpoint binds to a "private" socket, uses a "management" security realm, and enables administrative features.

```

<endpoints>
  <endpoint socket-binding="public"
    security-realm="application"
    admin="false">
    <hotrod-connector/>
    <rest-connector/>
  </endpoint>
  <endpoint socket-binding="private"
    security-realm="management">
    <hotrod-connector/>
    <rest-connector/>
  </endpoint>
</endpoints>

```



NOTE

As of 8.3 the **endpoints** element is now a wrapper for one or more **endpoint** configurations. If you add child elements to **endpoints** you must use the **endpoint** element.

If you are upgrading from Data Grid 8.2 or previous, and you configure multiple endpoints, you must adapt your configuration to the 8.3 schema. See [Migrating to Data Grid 8](#) for more information.

Security enhancements

Security realms for cluster transport

Specify a security realm that contains a TLS server identity to secure cluster transport, for example:

```

<cache-container>
  <transport server:security-realm="tls-transport"/>
</cache-container>

```



NOTE

This configuration also encrypts RELAY messages for cross-site replication and provides an alternative to the JGroups SYM_ENCRYPT and ASYM_ENCRYPT protocols. You must configure security on all Data Grid clusters that participate in cross-site replication.

PEM file keystores and trust stores

Add PEM files directly to Data Grid Server configuration to use them as trust stores and keystores in a TLS server identity.

- PEM files as keystores must contain a private key in PKCS#1 or PKCS#8 format and one or more certificates.
- PEM files as trust stores must contain one or more certificates.
- PEM files require an empty password, **password=""**.

Full support for TLSv1.3 with OpenSSL

Data Grid Server supports TLS version 1.2 and 1.3 by default. You can restrict the security protocol for client connections to TLSv1.3 only with the following configuration:

```
<engine enabled-protocols="TLSv1.3" />
```

Configuration overlays

Data Grid Server can parse multiple configuration files that you overlay on startup with repeated **--server-config=** or **-c** arguments.

```
$ bin/server.sh -c infinispan.xml -c overlay.yaml
```

You can use as many configuration overlay files as required, in any order. Configuration overlay files:

- Must each be valid Data Grid configuration.
- Do not need to be full configuration as long as your combination of overlay files results in a full configuration.



IMPORTANT

Data Grid Server does not detect conflicting configuration between overlay files. Each overlay file overwrites any conflicting configuration in the preceding configuration.

Cache configuration

If you add cache configuration to overlay files, Data Grid Server does not dynamically create that cache on other nodes.

Overlay files cannot contain cache configuration fragments. For example, to create distributed cache in an overlay file, **-c mycache.yaml**, you must include the **infinispan** and **cacheContainer** fields as follows:

```
infinispan:
  cacheContainer:
    distributedCache:
      name: "mycache"
      mode: "SYNC"
      owners: 2
      statistics: "true"
      encoding:
        mediaType: "application/x-protostream"
```

Listening on all addresses

If you specify the **0.0.0.0** meta-address (**INADDR_ANY**) as the bind address for Data Grid Server, it listens for incoming client connections on all available network interfaces.

Performance and usability improvements

- Data Grid Server 8.3 consolidates thread pools to use fewer host resources.
- Data Grid Server installations now include a **bin/service.bat** script for running as a Microsoft Windows service.

1.1.6. Data Grid Console

Data Grid Console is a graphical user interface for monitoring and managing remote caches that includes the following features and enhancements in 8.3:

- Caches must now have a suitable encoding to view the **Entries** tab.
- You can turn cluster rebalancing operations for caches on and off.



IMPORTANT

You should disable cluster rebalancing only for short periods of time to avoid repeated cluster rebalancing operations when restarting a node or adding multiple nodes. Always enable cluster rebalancing as soon as possible to prevent data loss.

- Cache templates are no longer available in the **Add cache** dialog unless you add custom templates to the Data Grid Server configuration.
- Automatic detection of JSON encoding and Protobuf custom types.
- Cache entry editing is now type-aware.
- Alert modals, tooltips, and inline help are updated for improved user experience.
- Improved error handling and statistics collection.

1.1.7. Data Grid Command Line Interface (CLI)

The Data Grid CLI lets you script and automate administrative operations for remote caches.

Modifying configuration attributes at runtime

The **alter** command lets you modify Data Grid configuration attributes at runtime.

- Use the **--file** option to specify a configuration file with one or more attribute modifications.
- Use the **--attribute** and **--value** option to modify a specific configuration attribute.

```
alter cache mycache --attribute=clustering.remote-timeout --value=5000
```

Controlling cache availability

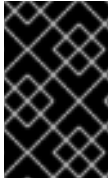
The **availability** command lets you manage availability of clustered caches in network partitions.

- Use the **--mode=[AVAILABLE|DEGRADED_MODE]** option to set cache availability to AVAILABLE or DEGRADED_MODE when using either the DENY_READ_WRITES or ALLOW_READS partition handling strategy.

```
availability --mode=AVAILABLE mycache
```

Enabling and disabling rebalancing operations

The **rebalance** command lets you turn cluster rebalancing operations for caches on and off.



IMPORTANT

You should disable cluster rebalancing only for short periods of time to avoid repeated cluster rebalancing operations when restarting a node or adding multiple nodes. Always enable cluster rebalancing as soon as possible to prevent data loss.

- Use the **rebalance disable** command to turn rebalancing off.

```
[//containers/default]> rebalance disable
```

- Use the **rebalance enable** command to turn rebalancing on.

```
[//containers/default]> rebalance enable
```

Hot Rod cluster migration

The **migrate** command provides additional options for migrating data from one running Data Grid cluster to another.

- Use the **migrate cluster connect** command to establish a connection from the source cluster to the target cluster.
- Use the **migrate cluster synchronize** command to migrate data from the source cluster to the target cluster.
- Use the **migrate cluster disconnect** command to end the connection between clusters after you migrate data.

Retrieving cross-site replication relay nodes

The **site** command lets you find out which nodes in the Data Grid cluster handle RELAY messages for cross-site replication.

- Use the **site is-relay-node** command to find out if a node is a relay node.
- Use the **site relay-nodes** command to get a list of relay nodes in the cluster.

Native CLI

The native CLI is updated in 8.3 to improve usage as an **oc** client plugin and for interacting with Data Grid pods running on Red Hat OpenShift.

1.1.8. REST API

Data Grid gives you access to remote caches and containers with HTTP over REST.

Modifying configuration attributes at runtime

- Retrieve all mutable cache configuration attributes for caches.

```
GET /rest/v2/caches/{cacheName}?action=get-mutable-attributes
```

- Change a mutable cache configuration attribute.

```
POST /rest/v2/caches/{cacheName}?action=set-mutable-attributes&attribute-name={attributeName}&attribute-value={attributeValue}
```

Controlling cache availability

- Retrieve the availability of a cache.

```
GET /v2/caches/{cacheName}?action=get-availability
```

- Change the availability of clustered caches when using either the DENY_READ_WRITES or ALLOW_READS partition handling strategy.

```
POST /v2/caches/{cacheName}?action=set-availability&availability={AVAILABILITY}
```

Enabling and disabling rebalancing operations

- Turn on automatic rebalancing for all caches.

```
POST /rest/v2/cache-managers/{cacheManagerName}?action=enable-rebalancing
```

- Turn off automatic rebalancing for all caches.

```
POST /rest/v2/cache-managers/{cacheManagerName}?action=disable-rebalancing
```

- Turn on automatic rebalancing for a specific cache.

```
POST /v2/caches/{cacheName}?action=enable-rebalancing
```

- Turn off automatic rebalancing for a specific cache.

```
POST /v2/caches/{cacheName}?action=disable-rebalancing
```

Hot Rod migration

The REST API exposes controls that let you migrate data from one running Data Grid cluster to another.

- Establish a connection from the source cluster to the target cluster.

```
POST /v2/caches/myCache?action=connect-source
```

- Migrate data from the source cluster to the target cluster.

```
POST /v2/caches/myCache?action=sync-data
```

- End the connection between clusters after you migrate data.

```
DELETE /v2/caches/myCache/rolling-upgrade/source-connection
```

Retrieving cross-site replication relay nodes

- Get the list of nodes that send and receive RELAY messages for cross-site replication:

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

Converting cache configuration between XML, JSON, and YAML

Invoke a **POST** request with valid configuration and the **?action=convert** parameter. Data Grid responds with the equivalent representation of the configuration in the type specified by the **Accept** header.

```
POST /rest/v2/caches?action=convert
```

Cache entry and cache configuration listeners with server-sent events

- Receive events about configuration changes using server-sent events.

```
GET /rest/v2/container/config?action=listen
```

- Receive cache events using server-sent events.

```
GET /rest/v2/caches/{name}?action=listen
```

1.1.9. Query API

Data Grid lets you index caches and perform relational or full-text queries with in the Ickle query language.

Delete by query

You can delete entries from Data Grid caches with the following syntax:

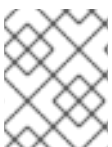
```
DELETE FROM <entityName> [WHERE condition]
```

- Reference only single entities with **<entityName>**. DELETE queries cannot use joins.
- WHERE conditions are optional.

Query execution methods

The **Query** API provides two methods for executing Ickle queries on a cache:

- **Query.execute()** runs a SELECT statement and returns a result.
- **Query.executeStatement()** runs a DELETE statement and modifies data.



NOTE

You should always invoke **executeStatement()** to modify data and invoke **execute()** to get the result of a query.

1.1.10. Cross-site replication

Cross-site replication lets you back up Data Grid clusters across multiple geographic regions.

Improved observability and metrics

Data Grid 8.3 updates cross-site replication capabilities to improve monitoring and statistics.

Data Grid lets you:

- Retrieve detailed status of backup locations via the CLI, REST API, or JMX.
- Retrieve the list of relay nodes for backup locations to find out which ones send JGroups RELAY messages for cross-site communication.



NOTE

JGroups configuration refers to relay nodes as "site master" nodes. Data Grid uses relay node instead because it is more descriptive and presents a more intuitive choice.

Data Grid 8.3 also provides more detailed metrics about cross-site replication operations for each backup location and for caches, including response times and number of RELAY messages.

Naming global clusters

The **cluster** attribute lets you define a name for global clusters that is printed to log messages. This makes it easier to distinguish global clusters from local clusters in Data Grid logs.



NOTE

Global cluster names that you specify with the **cluster** attribute must be the same at all sites.

By default cross-site clusters are named "xsite" but you can specify a custom name, such as "my-global-cluster" in the following example:

```
<remote-sites default-stack="tcp" cluster="my-global-cluster">
  <remote-site name="LON"/>
  <remote-site name="NYC"/>
</remote-sites>
```

1.1.11. Rolling upgrades and Hot Rod migration

Data Grid 8.3 simplifies the configuration and mechanisms for performing live migration of data across clusters to move from one version to another. Rolling upgrades in Data Grid 8.3:

- No longer need to manually add **remote-store** configuration to caches to perform rolling upgrades.
Use the CLI or REST API to connect source and target clusters.
- Provide additional methods in the REST API and CLI commands for rolling upgrade operations.
- Let you perform rolling upgrades with caches that you create dynamically through the Console, CLI, REST API, or Hot Rod API.

1.2. WHAT'S NEW IN DATA GRID 8.3.1

Data Grid 8.3 improves usability, increases performance, and enhances security. Find out what's new.

Compliance with the Federal Information Processing Standards (FIPS) 140-2

You can now install and run Data Grid Server on a RHEL 8 system with FIPS 140-2 compliant cryptography. For more information see [Configuring Data Grid Server on a system with FIPS 140-2 compliant cryptography](#).

Improved metrics

This release includes several enhancements to the Data Grid metrics endpoint.

Approximate cache size statistics

Data Grid 8.3.1 introduces statistics that provide approximate values for cache size. In 8.3.0, Data Grid disabled cache size statistics because the operation to calculate them negatively impacted performance.

The following statistics are enabled in the default metrics configuration:

approximateEntries

Approximate current number of entry replicas in the cache on the server that receives the request. Includes both entries in memory and in persistent storage.

approximateEntriesUnique

Approximate current number of entries for which the server that receives the request is the primary owner. Includes both entries in memory and in persistent storage.

globalApproximateEntries

Approximate current number of entry replicas currently in the cache cluster-wide. Includes both entries in memory and in persistent storage.

globalApproximateEntriesUnique

Approximate current number of unique entries in the cache cluster-wide. Includes both entries in memory and in persistent storage. Entries owned by multiple nodes are counted only once.

Remote site status metrics for cross-site connection

Data Grid now exposes metrics about the status of each backup location through the **metrics** endpoint. You can view whether a specific site is online or offline from any monitoring tool that supports the OpenMetrics format, such as Prometheus.

JGroups attributes exposed through the metrics endpoint

Data Grid lets you consume Java Management Extensions (JMX) statistics through the **metrics** endpoint. You can now monitor JGroups protocol events.

Performance enhancements to cross-site replication

This release includes several enhancements to cross-site replication capabilities that improve performance, including conflict detection and resolution for the asynchronous backup strategy.

Data Grid also lets you configure frequency for periodical cleanup of metadata, known as tombstones. You can configure the frequency for the cleanup task by defining a target size for tombstone maps as well as the maximum delay between task runs.

For more information see [Data Grid Cross-Site Replication](#).

1.3. SUPPORTED JAVA VERSIONS IN DATA GRID 8.3

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

Embedded caches

Red Hat supports Java 8, Java 11, and Java 17 when using Data Grid for embedded caches in custom applications.

Remote caches

Red Hat supports Java 11 and Java 17 for Data Grid Server installations. For Hot Rod Java clients, Red Hat supports Java 8, Java 11, and Java 17.



NOTE

When running Data Grid Server on bare metal installations, the JavaScript engine is not available with Java 17.

Java 8 deprecation

As of Data Grid 8.2, support for Java 8 is deprecated 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 8.3](#)
- [Data Grid Deprecated Features and Functionality](#)

CHAPTER 2. KNOWN AND FIXED ISSUES

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

2.1. KNOWN ISSUES FOR DATA GRID

For issues that affect Data Grid clusters that you manage with Data Grid Operator, you should refer to the [Data Grid Operator 8.3 release notes](#).

Deadlock occurs when `putAll()` operations write to expired entries with optimistic locking

Issue: [JDG-5087](#)

Description: With transactional caches that use optimistic locking, commands to remove expired entries acquire locks even when the expiration was triggered by a write operation. In some cases, when `putAll()` operations write to expired entries, this behavior can lead to deadlocks.

Workaround: There is no workaround for this issue.

Inconsistent transactions when network partitions occur

Issue: [JDG-3935](#)

Description: In scenarios where a network partition occurs for a Data Grid cluster, transactions are rolled back after the partition is healed.

Workaround: There is no workaround for this issue.

Session externalization from Red Hat JBoss Web Server to Data Grid 8.3 is available with 7.3.8 or 8.1.1 versions of the Tomcat session client

Issue: [JDG-4599](#)

Description: Data Grid 8.3 does not yet include the Tomcat session client, which will be available after EAP 7.4 GA.

Workaround: Use Data Grid Server 8.3 in combination with the Data Grid 8.1.1 version of the Tomcat session client with the following configuration:

```
<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"/>
```

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 Server cannot use some credential keystores

Issue: [JDG-4711](#)

Description: In some cases, Data Grid Server logs the following exceptions when using credential keystores:

```
java.lang.RuntimeException: org.wildfly.security.credential.store.CredentialStoreException:
ELY09508: Cannot write credential to store
...
Caused by: org.wildfly.security.credential.store.CredentialStoreException: ELY09508: Cannot write
credential to store
...
Caused by: java.security.KeyStoreException: Cannot store non-PrivateKeys
```

In most cases, this error indicates that a JKS keystore contains symmetric keys instead of public and private keys.

Workaround: Ensure that the credential keystore uses asymmetric keys.

2.2. FIXED IN DATA GRID 8.3.0

Data Grid 8.3.0 includes the following notable fixes:

- [JDG-4947](#) Asynchronous cross-site replication results in tombstone leaks
- [JDG-5028](#) Data race occurs when handling expired entries with `putAll()` or `getAll()` operations
- [JDG-4763](#) Clients cannot connect to remote caches that use TLS/SSL encryption
- [JDG-4797](#) Hot Rod client manual cluster switch not working
- [JDG-4860](#) Hot Rod Java client retries operations too many times
- [JDG-3629](#) Maximum idle expiration does not work with eviction and cache stores configured with passivation
- [JDG-3916](#) Data Grid Console logs incomplete messages when turning sites on or off
- [JDG-4530](#) Cross-site replication automatic state transfer with empty exception using set state mode
- [JDG-4598](#) Data Grid Console prompts users to select a cache template when cache configuration is not valid
- [JDG-4852](#) Data Grid Server running as a Microsoft Windows service does not shutdown if Hot Rod clients are connected

2.3. FIXED IN DATA GRID 8.3.1

Data Grid 8.3.1 includes the following notable fixes:

- [JDG-2796](#) Data Grid does not passivate JWS sessions correctly

- [JDG-4092](#) Cache persistent state is not deleted after successful cluster restart
- [JDG-5220](#) Ickle fulltext query not working with query parameters
- [JDG-5097](#) Backups fail when a Protobuf schema key contains forward slashes
- [JDG-5076](#) RemoveExpiredCommand does not remove entries on backup if equals is not implemented
- [JDG-5104](#) Data Grid Modules for Red Hat JBoss EAP (EAP) are missing dependencies
- [JDG-4657](#) Hot Rod client fails to connect when there is no bind address argument and servers are not on the same subnet
- [JDG-5245](#) ClassCastException with Oracle string-keyed-jdbc-store
- [JDG-5161](#) Compactor fails when removing content from persistent storage

2.4. HOST SYSTEM AND DEPENDENCY ISSUES

In some cases Data Grid deployments can encounter errors that are caused by the host system or external dependency. This section provides details about any such known issues as well as troubleshooting and workaround procedures.

TLS on Red Hat Enterprise Linux 7

RHEL 7 provides a version of the OpenSSL library that does not yet offer support for TLSv1.3. However Data Grid Server 8.3 enables TLSv1.3 and TLSv1.2 by default, which causes errors with client connections for encrypted Hot Rod and REST endpoints.

Data Grid Server also logs messages such as the following:

```
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.
```

If you install Data Grid Server on RHEL 7 you should use the native Java SSL library by disabling OpenSSL with the following JVM option:

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

Additional resources

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