



# Red Hat xPaaS

# 0

# Red Hat xPaaS Data Grid Image

---

Using the Data Grid xPaaS Image

Red Hat xPaaS Documentation  
Team



Red Hat xPaaS 0 Red Hat xPaaS Data Grid Image

---

Using the Data Grid xPaaS Image

## Legal Notice

Copyright © 2016 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, 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 Software Collections 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

Guide to using the Red Hat xPaaS Data Grid image

---

## Table of Contents

<b>CHAPTER 1. INTRODUCTION TO THE DATA GRID IMAGE</b> .....	<b>3</b>
<b>CHAPTER 2. BEFORE YOU BEGIN</b> .....	<b>4</b>
2.1. FUNCTIONALITY DIFFERENCES FOR OPENSIFT JBOSS DATA GRID XPAAS IMAGES	4
2.2. INITIAL SETUP	4
2.3. FORMING A CLUSTER USING THE OPENSIFT JBOSS DATA GRID XPAAS IMAGES	4
2.4. ENDPOINTS	5
2.5. CONFIGURING CACHES	5
2.6. DATASOURCES	5
2.7. SECURITY DOMAINS	7
2.8. MANAGING OPENSIFT JBOSS DATA GRID XPAAS IMAGES	7
<b>CHAPTER 3. GET STARTED</b> .....	<b>8</b>
3.1. USING THE JBOSS DATA GRID XPAAS IMAGE SOURCE-TO-IMAGE (S2I) PROCESS	8
3.2. USING A MODIFIED JBOSS DATA GRID XPAAS IMAGE	9
<b>CHAPTER 4. REFERENCE</b> .....	<b>10</b>
4.1. INFORMATION ENVIRONMENT VARIABLES	10
4.2. CONFIGURATION ENVIRONMENT VARIABLES	10
4.3. CACHE ENVIRONMENT VARIABLES	14
4.4. DATASOURCE ENVIRONMENT VARIABLES	17
4.5. SECURITY ENVIRONMENT VARIABLES	18
4.6. EXPOSED PORTS	19
4.7. TROUBLESHOOTING	20



## CHAPTER 1. INTRODUCTION TO THE DATA GRID IMAGE

Red Hat JBoss Data Grid is available as a containerized xPaaS image that is designed for use with OpenShift. This image provides an in-memory distributed database so that developers can quickly access large amounts of data in a hybrid environment.



### Important

There are significant differences in supported configurations and functionality in the JBoss Data Grid xPaaS image compared to the full, non-PaaS release of JBoss Data Grid.

This topic details the differences between the JBoss Data Grid xPaaS image and the full, non-PaaS release of JBoss Data Grid, and provides instructions specific to running and configuring the JBoss Data Grid xPaaS image. Documentation for other JBoss Data Grid functionality not specific to the JBoss Data Grid xPaaS image can be found in the [JBoss Data Grid documentation on the Red Hat Customer Portal](#).

## CHAPTER 2. BEFORE YOU BEGIN

### 2.1. FUNCTIONALITY DIFFERENCES FOR OPENSIFT JBOSS DATA GRID XPAAS IMAGES

There are several major functionality differences in the OpenShift JBoss Data Grid xPaaS image:

- ✦ The JBoss Data Grid Management Console is not available [to manage OpenShift JBoss Data Grid xPaaS images](#).
- ✦ The JBoss Data Grid Management CLI is only bound locally. This means that you can only access the Management CLI of a container from within the pod.
- ✦ Library mode is not supported.
- ✦ Only JDBC is supported for a backing cache-store. Support for remote cache stores are present only for data migration purposes.

### 2.2. INITIAL SETUP

The instructions in this guide follow on from and assume an OpenShift instance similar to that created in the [OpenShift Primer](#).

### 2.3. FORMING A CLUSTER USING THE OPENSIFT JBOSS DATA GRID XPAAS IMAGES

Clustering is achieved through one of two discovery mechanisms: Kubernetes or DNS. This is accomplished by configuring the JGroups protocol stack in *clustered-openshift.xml* with either the `<openshift.KUBE_PING/>` or `<openshift.DNS_PING/>` elements. By default *KUBE\_PING* is the pre-configured and supported protocol.

For *KUBE\_PING* to work the following steps must be taken:

1. The *OPENSIFT\_KUBE\_PING\_NAMESPACE* environment variable must be set (as seen in the [Configuration Environment Variables](#)). If this variable is not set, then the server will act as if it is a single-node cluster, or a cluster that consists of only one node.
2. The *OPENSIFT\_KUBE\_PING\_LABELS* environment variable must be set (as seen in the [Configuration Environment Variables](#)). If this variable is not set, then pods outside the application (but in the same namespace) will attempt to join.
3. Authorization must be granted to the service account the pod is running under to be allowed to Kubernetes' REST api. This is done on the command line:

#### Example 2.1. Policy commands

Using the *default* service account in the *myproject* namespace:

```
oc policy add-role-to-user view system:serviceaccount:$(oc project -q):default -n $(oc project -q)
```



Using the **eap-service-account** in the **myproject** namespace:

```
oc policy add-role-to-user view system:serviceaccount:$(oc
project -q):eap-service-account -n $(oc project -q)
```

Once the above is configured images will automatically join the cluster as they are deployed; however, removing images from an active cluster, and therefore shrinking the cluster, is not supported.

## 2.4. ENDPOINTS

Clients can access JBoss Data Grid via REST, HotRod, and memcached endpoints defined as usual in the cache's configuration.

If a client attempts to access a cache via HotRod and is in the same project it will be able to receive the full cluster view and make use of consistent hashing; however, if it is in another project then the client will be unable to receive the cluster view. Additionally, if the client is located outside of the project that contains the HotRod cache there will be additional latency due to extra network hops being required to access the cache.



### Important

Only caches with an exposed REST endpoint will be accessible outside of OpenShift.

## 2.5. CONFIGURING CACHES

A list of caches may be defined by the **CACHE\_NAMES** environment variable. By default the following caches are created:

- ✦ **default**
- ✦ **memcached**

Each cache's behavior may be controlled through the use of cache-specific environment variables, with each environment variable expecting the cache's name as the prefix. For instance, consider the **default** cache, any configuration applied to this cache must begin with the **DEFAULT\_** prefix. To define the number of cache entry owners for each entry in this cache the **DEFAULT\_CACHE\_OWNERS** environment variable would be used.

A full list of these is found at [Cache Environment Variables](#).

## 2.6. DATASOURCES

Datasources are automatically created based on the value of some environment variables.

The most important variable is the **DB\_SERVICE\_PREFIX\_MAPPING** which defines JNDI mappings for datasources. It must be set to a comma-separated list of **<name><database\_type>=<PREFIX> triplet**, where **\*name** is used as the pool-name in the datasource, **database\_type** determines which database driver to use, and **PREFIX** is the prefix used in the names of environment variables, which are used to configure the datasource.

### 2.6.1. JNDI Mappings for Datasources

For each `<name>-database_type=<PREFIX>` triplet in the `DB_SERVICE_PREFIX_MAPPING` environment variable, a separate datasource will be created by the launch script, which is executed when running the image.

The `<database_type>` will determine the driver for the datasource. Currently, only `postgresql` and `mysql` are supported.

The `<name>` parameter can be chosen on your own. Do not use any special characters.



#### Note

The first part (before the equal sign) of the `DB_SERVICE_PREFIX_MAPPING` should be lowercase.

### 2.6.2. Database Drivers

The JBoss Data Grid xPaaS image contains Java drivers for MySQL, PostgreSQL, and MongoDB databases deployed. Datasources are **generated only for MySQL and PostgreSQL databases**.



#### Note

For MongoDB databases there are no JNDI mappings created because this is not a SQL database.

### 2.6.3. Examples

The following examples demonstrate how datasources may be defined using the `DB_SERVICE_PREFIX_MAPPING` environment variable.

#### 2.6.3.1. Single Mapping

Consider the value `test-postgresql=TEST`.

This will create a datasource named `java:jboss/datasources/test_postgresql`. Additionally, all of the required settings, such as username and password, will be expected to be provided as environment variables with the `TEST_` prefix, such as `TEST_USERNAME` and `TEST_PASSWORD`.

#### 2.6.3.2. Multiple Mappings

Multiple database mappings may also be specified; for instance, considering the following value for the `DB_SERVICE_PREFIX_MAPPING` environment variable: `cloud-postgresql=CLOUD,test-mysql=TEST_MYSQL`.



#### Note

Multiple datasource mappings should be separated with commas, as seen in the above example.

This will create two datasources:

1. `java:jboss/datasources/test_mysql`
2. `java:jboss/datasources/cloud_postgresql`

MySQL datasource configuration, such as the username and password, will be expected with the `TEST_MYSQL` prefix, for example `TEST_MYSQL_USERNAME`. Similarly the PostgreSQL datasource will expect to have environment variables defined with the `CLOUD_` prefix, such as `CLOUD_USERNAME`.

## 2.6.4. Environment Variables

A full list of datasource environment variables may be found at [Datasource Environment Variables](#).

## 2.7. SECURITY DOMAINS

To configure a new Security Domain the `SECDOMAIN_NAME` environment variable must be defined, which will result in the creation of a security domain named after the passed in value. This domain may be configured through the use of the [Security Environment Variables](#).

## 2.8. MANAGING OPENSIFT JBOSS DATA GRID XPAAS IMAGES

A major difference in managing an OpenShift JBoss Data Grid xPaaS image is that there is no Management Console exposed for the JBoss Data Grid installation inside the image. Because images are intended to be immutable, with modifications being written to a non-persistent file system, the Management Console is not exposed.

However, the JBoss Data Grid Management CLI (`JDG_HOME/bin/jboss-cli.sh`) is still accessible from within the container for troubleshooting purposes.

1. First open a remote shell session to the running pod:

```
$ oc rsh <pod_name>
```

2. Then run the following from the remote shell session to launch the JBoss Data Grid Management CLI:

```
$ /opt/datagrid/bin/jboss-cli.sh
```

### Warning

Any configuration changes made using the JBoss Data Grid Management CLI on a running container will be lost when the container restarts.

Making configuration changes to the JBoss Data Grid instance inside the JBoss Data Grid xPaaS image is different from the process you may be used to for a regular release of JBoss Data Grid.

## CHAPTER 3. GET STARTED

The Red Hat xPaaS middleware images were [automatically created during the installation](#) of OpenShift along with the other default image streams and templates.

You can make changes to the JBoss Data Grid configuration in the xPaaS image using either the S2I templates, or by using a modified JBoss Data Grid xPaaS image.

### 3.1. USING THE JBOSS DATA GRID XPAAS IMAGE SOURCE-TO-IMAGE (S2I) PROCESS

The recommended method to run and configure the OpenShift JBoss Data Grid xPaaS image is to use the OpenShift S2I process together with the application template parameters and environment variables.

The S2I process for the JBoss Data Grid xPaaS image works as follows:

1. If there is a *pom.xml* file in the source repository, a Maven build is triggered with the contents of `$MAVEN_ARGS` environment variable.
2. By default the `package` goal is used with the `openshift` profile, including the system properties for skipping tests (`-DskipTests`) and enabling the Red Hat GA repository (`-Dcom.redhat.xpaas.repo.redhatga`).
3. The results of a successful Maven build are copied to `JDG_HOME/standalone/deployments`. This includes all JAR, WAR, and EAR files from the directory within the source repository specified by `$ARTIFACT_DIR` environment variable. The default value of `$ARTIFACT_DIR` is the `target` directory.
  - ✦ Any JAR, WAR, and EAR in the `deployments` source repository directory are copied to the `JDG_HOME/standalone/deployments` directory.
  - ✦ All files in the `configuration` source repository directory are copied to `JDG_HOME/standalone/configuration`.



#### Note

If you want to use a custom JBoss Data Grid configuration file, it should be named `clustered-openshift.xml`.

4. All files in the `modules` source repository directory are copied to `JDG_HOME/modules`.

#### 3.1.1. Using a Different JDK Version in the JBoss Data Grid xPaaS Image

The JBoss Data Grid xPaaS image may come with multiple versions of OpenJDK installed, but only one is the default. For example, the JBoss Data Grid 6.5 xPaaS image comes with OpenJDK 1.7 and 1.8 installed, but OpenJDK 1.8 is the default.

If you want the JBoss Data Grid xPaaS image to use a different JDK version than the default, you must:

- ✦ Ensure that your *pom.xml* specifies to build your code using the intended JDK version.

- ✦ In the S2I application template, configure the image's **JAVA\_HOME** environment variable to point to the intended JDK version. For example:

```
{  
  "name": "JAVA_HOME",  
  "value": "/usr/lib/jvm/java-1.7.0"  
}
```

## 3.2. USING A MODIFIED JBOSS DATA GRID XPAAS IMAGE

An alternative method is to make changes to the image, and then use that modified image in OpenShift.

The JBoss Data Grid configuration file that OpenShift uses inside the JBoss Data Grid xPaaS image is ***JDG\_HOME/standalone/configuration/clustered-openshift.xml***, and the JBoss Data Grid startup script is ***JDG\_HOME/bin/openshift-launch.sh***.

You can run the JBoss Data Grid xPaaS image in Docker, make the required configuration changes using the JBoss Data Grid Management CLI (***JDG\_HOME/bin/jboss-cli.sh***), and then commit the changed container as a new image. You can then use that modified image in OpenShift.



### Important

It is recommended that you do not replace the OpenShift placeholders in the JBoss Data Grid xPaaS configuration file, as they are used to automatically configure services (such as messaging, datastores, HTTPS) during a container's deployment. These configuration values are intended to be set using environment variables.



### Note

Ensure that you follow the [guidelines for creating images](#).

## CHAPTER 4. REFERENCE

### 4.1. INFORMATION ENVIRONMENT VARIABLES

The following information environment variables are designed to convey information about the image and should not be modified by the user:

Table 4.1. Information Environment Variables

Variable Name	Description	Value
<b><i>JBOSS_DATAGRID_VERSION</i></b>	The full, non-PaaS release that the xPaaS image is based from.	<b><i>6.5.1.GA</i></b>
<b><i>JBOSS_HOME</i></b>	The directory where the JBoss distribution is located.	<b><i>/opt/datagrid</i></b>
<b><i>JBOSS_IMAGE_NAME</i></b>	Image name, same as <b><i>Name</i></b> label	<b><i>jboss-datagrid-6/datagrid65-openshift</i></b>
<b><i>JBOSS_IMAGE_RELEASE</i></b>	Image release, same as <b><i>Release</i></b> label	Example: dev
<b><i>JBOSS_IMAGE_VERSION</i></b>	Image version, same as <b><i>Version</i></b> label	Example: <b><i>1.2</i></b>
<b><i>JBOSS_MODULES_SYSTEM_PKGS</i></b>		<b><i>org.jboss.logmanager</i></b>
<b><i>JBOSS_PRODUCT</i></b>		<b><i>datagrid</i></b>
<b><i>LAUNCH_JBOSS_IN_BACKG ROUND</i></b>	Allows the data grid server to be gracefully shutdown even when there is no terminal attached.	<b><i>true</i></b>

### 4.2. CONFIGURATION ENVIRONMENT VARIABLES

Configuration environment variables are designed to conveniently adjust the image without requiring a rebuild, and should be set by the user as desired.

Table 4.2. Configuration Environment Variables

Variable Name	Description	Value
<b>CACHE_CONTAINER_START</b>	Should this cache container be started on server startup, or lazily when requested by a service or deployment. Defaults to <b>LAZY</b>	Example: <b>EAGER</b>
<b>CACHE_CONTAINER_STATISTICS</b>	Determines if the cache container collects statistics. Disable for optimal performance. Defaults to <b>true</b> .	Example: <b>false</b>
<b>CACHE_NAMES</b>	List of caches to configure. Defaults to <b>default,memcached</b> , and each defined cache will be configured as a distributed-cache with a mode of <b>SYNC</b> .	Example: <b>addressbook,addressbook_indexed</b>
<b>CONTAINER_SECURITY_CUSTOM_ROLE_MAPPER_CLASS</b>	Class of the custom principal to role mapper.	Example: <b>com.acme.CustomRoleMapper</b>
<b>CONTAINER_SECURITY_IDENTITY_ROLE_MAPPER</b>	Set a role mapper for this cache container. Valid values are: <b>identity-role-mapper,common-name-role-mapper,cluster-role-mapper,custom-role-mapper</b> .	Example: <b>identity-role-mapper</b>
<b>CONTAINER_SECURITY_ROLES</b>	Define role names and assign permissions to them.	Example: <b>admin=ALL,reader=READ,writer=WRITE</b>
<b>DB_SERVICE_PREFIX_MAPPING</b>	Define a comma-separated list of datasources to configure.	Example: <b>test-mysql=TEST_MYSQL</b>
<b>DEFAULT_CACHE</b>	Indicates the default cache for this cache container.	Example: <b>addressbook</b>

Variable Name	Description	Value
<b><i>ENCRYPTION_REQUIRE_SSL_CLIENT_AUTH</i></b>	Whether to require client certificate authentication. Defaults to <b><i>false</i></b> .	Example: <b><i>true</i></b>
<b><i>HOTROD_AUTHENTICATION</i></b>	If defined the hotrod-connectors will be configured with authentication in the <b><i>ApplicationRealm</i></b> .	Example: <b><i>true</i></b>
<b><i>HOTROD_ENCRYPTION</i></b>	If defined the hotrod-connectors will be configured with encryption in the <b><i>ApplicationRealm</i></b> .	Example: <b><i>true</i></b>
<b><i>HOTROD_SERVICE_NAME</i></b>	Name of the OpenShift service used to expose HotRod externally.	Example: <b><i>DATAGRID_APP_HOTROD</i></b>
<b><i>INFINISPAN_CONNECTORS</i></b>	Comma separated list of connectors to configure. Defaults to <b><i>hotrod,memcached,rest</i></b> . Note that if authorization or authentication is enabled on the cache then memcached should be removed as this protocol is inherently insecure.	Example: <b><i>hotrod</i></b>
<b><i>JAVA_OPTS_APPEND</i></b>	The contents of <b><i>JAVA_OPTS_APPEND</i></b> is appended to <b><i>JAVA_OPTS</i></b> on startup.	Example: <b><i>-Dfoo=bar</i></b>
<b><i>JGROUPS_CLUSTER_PASSWORD</i></b>	A password to control access to JGroups. Needs to be set consistently cluster-wide. The image default is to use the <b><i>OPENSIFT_KUBE_PING_LABELS</i></b> variable value; however, the JBoss application templates generate and supply a random value.	Example: <b><i>miR0JaDR</i></b>



Variable Name	Description	Value
<b>MEMCACHED_CACHE</b>	The name of the cache to use for the Memcached connector.	Example: <i>memcached</i>
<b>OPENSIFT_KUBE_PING_LABELS</b>	Clustering labels selector.	Example: <i>application=eap-app</i>
<b>OPENSIFT_KUBE_PING_NAMESPACE</b>	Clustering project namespace.	Example: <i>myproject</i>
<b>PASSWORD</b>	Password for the JDG user.	Example: <i>p@ssw0rd</i>
<b>REST_SECURITY_DOMAIN</b>	The security domain to use for authentication and authorization purposes. Defaults to <i>none</i> (no authentication).	Example: <i>other</i>
<b>TRANSPORT_LOCK_TIMEOUT</b>	Infinispan uses a distributed lock to maintain a coherent transaction log during state transfer or rehashing, which means that only one cache can be doing state transfer or rehashing at the same time. This constraint is in place because more than one cache could be involved in a transaction. This timeout controls the time to wait to acquire a distributed lock. Defaults to <i>240000</i> .	Example: <i>120000</i>
<b>USERNAME</b>	Username for the JDG user.	Example: <i>openshift</i>

### 4.3. CACHE ENVIRONMENT VARIABLES

The following environment variables all control behavior of individual caches; when defining these values for a particular cache substitute the cache's name for **CACHE\_NAME**.

Table 4.3. Cache Environment Variables

Variable Name	Description	Example Value
<b>&lt;CACHE_NAME&gt;_CACHE_TY PE</b>	Determines whether this cache should be distributed or replicated. Defaults to <b><i>distributed</i></b> .	<b><i>replicated</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_ST ART</b>	Determines if this cache should be started on server startup, or lazily when requested by a service or deployment. Defaults to <b><i>LAZY</i></b> .	<b><i>EAGER</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_BA TCHING</b>	Enables invocation batching for this cache. Defaults to <b><i>false</i></b> .	<b><i>true</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_ST ATISTICS</b>	Determines whether or not the cache collects statistics. Disable for optimal performance. Defaults to <b><i>true</i></b> .	<b><i>false</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_MO DE</b>	Sets the clustered cache mode, <b><i>ASYNC</i></b> for asynchronous operations, or <b><i>SYNC</i></b> for synchronous operations.	<b><i>ASYNC</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_QU EUE_SIZE</b>	In <b><i>ASYNC</i></b> mode this attribute can be used to trigger flushing of the queue when it reaches a specific threshold. Defaults to <b><i>0</i></b> , which disables flushing.	<b><i>100</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_QU EUE_FLUSH_INTERVAL</b>	In <b><i>ASYNC</i></b> mode this attribute controls how often the asynchronous thread runs to flush the replication queue. This should be a positive integer that represents thread wakeup time in milliseconds. Defaults to <b><i>10</i></b> .	<b><i>20</i></b>

Variable Name	Description	Example Value
<b>&lt;CACHE_NAME&gt;_CACHE_REMOTE_TIMEOUT</b>	In <b>SYNC</b> mode the timeout, in milliseconds, used to wait for an acknowledgement when making a remote call, after which the call is aborted and an exception is thrown. Defaults to <b>17500</b> .	<b>25000</b>
<b>&lt;CACHE_NAME&gt;_CACHE_OWNERS</b>	Number of cluster-wide replicas for each cache entry. Defaults to <b>2</b> .	<b>5</b>
<b>&lt;CACHE_NAME&gt;_CACHE_SEGMENTS</b>	Number of hash space segments per cluster. The recommended value is 10 * cluster size. Defaults to <b>80</b> .	<b>30</b>
<b>&lt;CACHE_NAME&gt;_CACHE_L1_LIFESPAN</b>	Maximum lifespan, in milliseconds, of an entry placed in the L1 cache. Defaults to <b>0</b> , indicating that L1 is disabled.	<b>100</b> .
<b>&lt;CACHE_NAME&gt;_CACHE_EVICTON_STRATEGY</b>	Sets the cache eviction strategy. Available options are <b>UNORDERED</b> , <b>FIFO</b> , <b>LRU</b> , <b>LIRS</b> , and <b>NONE</b> (to disable eviction). Defaults to <b>NONE</b> .	<b>FIFO</b>
<b>&lt;CACHE_NAME&gt;_CACHE_EVICTON_MAX_ENTRIES</b>	Maximum number of entries in a cache instance. If selected value is not a power of two the actual value will default to the least power of two larger than the selected value. A value of <b>-1</b> indicates no limit. Defaults to <b>10000</b> .	<b>-1</b>
<b>&lt;CACHE_NAME&gt;_CACHE_EXPIRATION_LIFESPAN</b>	Maximum lifespan, in milliseconds, of a cache entry, after which the entry is expired cluster-wide. Defaults to <b>-1</b> , indicating that the entries never expire.	<b>10000</b>

Variable Name	Description	Example Value
<b>&lt;CACHE_NAME&gt;_CACHE_EXPIRATION_MAX_IDLE</b>	Maximum idle time, in milliseconds, a cache entry will be maintained in the cache. If the idle time is exceeded, then the entry will be expired cluster-wide. Defaults to <b>-1</b> , indicating that the entries never expire.	<b>10000</b>
<b>&lt;CACHE_NAME&gt;_CACHE_EXPIRATION_INTERVAL</b>	Interval, in milliseconds, between subsequent runs to purge expired entries from memory and any cache stores. If you wish to disable the periodic eviction process altogether, then set the interval to <b>-1</b> . Defaults to <b>5000</b> .	<b>-1</b>
<b>&lt;CACHE_NAME&gt;_CACHE_COMPATIBILITY_ENABLED</b>	Enables compatibility mode for this cache. Disabled by default.	<b>true</b>
<b>&lt;CACHE_NAME&gt;_CACHE_COMPATIBILITY_MARSHALLER</b>	A marshaller to use for compatibility conversions.	<b>com.acme.CustomMarshaller</b>
<b>&lt;CACHE_NAME&gt;_JDBC_STORE_TYPE</b>	Type of JDBC store to configure. This value may either be <b>string</b> or <b>binary</b> .	<b>string</b>
<b>&lt;CACHE_NAME&gt;_JDBC_STORE_DATASOURCE</b>	Defines the jndiname of the datasource.	<b>java:jboss/datasources/ExampleDS</b>
<b>&lt;CACHE_NAME&gt;_KEYED_TABLE_PREFIX</b>	Defines the prefix prepended to the cache name used when composing the name of the cache entry table. Defaults to <b>ispn_entry</b> .	<b>JDG</b>
<b>&lt;CACHE_NAME&gt;_CACHE_INDEX</b>	The indexing mode of the cache. Valid values are <b>NONE</b> , <b>LOCAL</b> , and <b>ALL</b> . Defaults to <b>NONE</b> .	<b>ALL</b>

Variable Name	Description	Example Value
<b>&lt;CACHE_NAME&gt;_CACHE_INDEXING_PROPERTIES</b>	Comma separated list of properties to pass on to the indexing system.	<b><i>default.directory_provider=ram</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_SECURITY_AUTHORIZATION_ENABLED</b>	Enables authorization checks for this cache. Defaults to <b><i>false</i></b> .	<b><i>true</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_SECURITY_AUTHORIZATION_ROLES</b>	Sets the valid roles required to access this cache.	<b><i>admin,reader,writer</i></b>
<b>&lt;CACHE_NAME&gt;_CACHE_PARTITION_HANDLING_ENABLED</b>	If enabled, then the cache will enter degraded mode when it loses too many nodes. Defaults to <b><i>true</i></b> .	<b><i>false</i></b>

## 4.4. DATASOURCE ENVIRONMENT VARIABLES

Datasource properties may be configured with the following environment variables:

Table 4.4. Datasource Environment Variables

Variable Name	Description	Example Value
<b>&lt;NAME&gt;&lt;DATABASE_TYPE&gt;_SERVICE_HOST_</b>	Defines the database server's hostname or IP to be used in the datasource's <b><i>connection_url</i></b> property.	<b><i>192.168.1.3</i></b>
<b>&lt;NAME&gt;_DATABASE_TYPE&gt;_SERVICE_PORT</b>	Defines the database server's port for the datasource.	<b><i>5432</i></b>

Variable Name	Description	Example Value
<b>&lt;PREFIX&gt;_JNDI</b>	Defines the JNDI name for the datasource. Defaults to <b>java:jboss/datasources/&lt;name&gt;&lt;database_type&gt;</b> , where <b>name</b> and <b>database_type</b> are taken from the triplet definition. This setting is useful if you want to override the default generated JNDI name.	<b>java:jboss/datasources/test-postgresql</b>
<b>&lt;PREFIX&gt;_USERNAME</b>	Defines the username for the datasource.	<b>admin</b>
<b>&lt;PREFIX&gt;_PASSWORD</b>	Defines the password for the datasource.	<b>password</b>
<b>&lt;PREFIX&gt;_DATABASE</b>	Defines the database name for the datasource.	<b>myDatabase</b>
<b>&lt;PREFIX&gt;_TX_ISOLATION</b>	Defines the java.sql.Connection transaction isolation level for the database.	<b>TRANSACTION_READ_UNCOMMITTED</b>
<b>&lt;PREFIX&gt;_TX_MIN_POOL_SIZE</b>	Defines the minimum pool size option for the datasource.	<b>1</b>
<b>&lt;PREFIX&gt;_TX_MAX_POOL_SIZE</b>	Defines the maximum pool size option for the datasource.	<b>20</b>

## 4.5. SECURITY ENVIRONMENT VARIABLES

The following environment variables may be defined to customize the environment's security domain:

Table 4.5. Security Environment Variables

Variable Name	Description	Example Value
<b>SECDOMAIN_NAME</b>	Define in order to enable the definition of an additional security domain.	<b><i>myDomain</i></b>
<b>SECDOMAIN_PASSWORD_STACKING</b>	If defined, the password-stacking module option is enabled and set to the value <b><i>useFirstPass</i></b> .	<b><i>true</i></b>
<b>SECDOMAIN_LOGIN_MODULE</b>	The login module to be used. Defaults to <b><i>UsersRoles</i></b> .	<b><i>UsersRoles</i></b>
<b>SECDOMAIN_USERS_PROPERTIES</b>	The name of the properties file containing user definitions. Defaults to <b><i>users.properties</i></b> .	<b><i>users.properties</i></b>
<b>SECDOMAIN_ROLES_PROPERTIES</b>	The name of the properties file containing role definitions. Defaults to <b><i>roles.properties</i></b> .	<b><i>roles.properties</i></b>

## 4.6. EXPOSED PORTS

The following ports are exposed by default in the JBoss Data Grid xPaaS Image:

Value	Description
8443	Secure Web
8778	-
11211	memcached
11222	internal hotrod
11333	external hotrod

**Important**

The external hotrod connector is only available if the *HOTROD\_SERVICE\_NAME* environment variable has been defined.

## 4.7. TROUBLESHOOTING

In addition to viewing the OpenShift logs, you can troubleshoot a running JBoss Data Grid xPaaS Image container by viewing its logs. These are outputted to the container's standard out, and are accessible with the following command:

```
$ oc logs -f <pod_name> <container_name>
```

**Note**

By default, the OpenShift JBoss Data Grid xPaaS Image does not have a file log handler configured. Logs are only sent to the container's standard out.