



# Red Hat Data Grid 7.3

## Running the Data Grid Operator

Data Grid Documentation





## Legal Notice

Copyright © 2019 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

Create and manage Red Hat Data Grid clusters running on Red Hat OpenShift.

---

## Table of Contents

<b>PREFACE</b> .....	<b>3</b>
<b>CHAPTER 1. TECHNOLOGY PREVIEW</b> .....	<b>4</b>
<b>CHAPTER 2. SPINNING UP DATA GRID CLUSTERS</b> .....	<b>5</b>
2.1. CREATING MINIMAL DATA GRID CLUSTERS	5
2.2. CREATING DATA GRID CLUSTERS WITH CREDENTIALS	5
2.3. VERIFYING DATA GRID CLUSTERS	7
<b>CHAPTER 3. MANAGING CLUSTER CREDENTIALS</b> .....	<b>9</b>
3.1. RETRIEVING CLUSTER CREDENTIALS	9



## PREFACE

The Data Grid Operator provides operational intelligence and reduces management complexity for deploying Data Grid on OpenShift clusters.

## CHAPTER 1. TECHNOLOGY PREVIEW

The Data Grid Operator is currently a Technology Preview.



### IMPORTANT

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.

For more information see [Red Hat Technology Preview Features Support Scope](#) .



## CHAPTER 2. SPINNING UP DATA GRID CLUSTERS

You create Data Grid clusters from custom resource definitions.

### Prerequisites

Install the Data Grid Operator from the OperatorHub.

## 2.1. CREATING MINIMAL DATA GRID CLUSTERS

Quickly spin up a Data Grid cluster with two nodes and let the Data Grid Operator generate credentials. You can retrieve the credentials from the pods after you create the Data Grid cluster. However, because Data Grid clusters require authentication, you cannot connect to the pods without the credentials.

### Procedure

1. Create a custom resource **yaml** for the minimal cluster.

```
$ cat > cr_minimal.yaml<<EOF
apiVersion: infinispn.org/v1
kind: Infinispn
metadata:
  name: example-rhdg
spec:
  replicas: 2
EOF
```

- **replicas** specifies the number of nodes in the Data Grid cluster.

2. Apply the custom resource **yaml**.

```
$ oc apply -f cr_minimal.yaml
```

3. Verify that the Data Grid Operator creates the pods.

```
$ oc get pods -w
```

NAME	READY	STATUS	RESTARTS	AGE
example-rhdg-1	0/1	ContainerCreating	0	4s
example-rhdg-2	0/1	ContainerCreating	0	4s
example-rhdg-3	0/1	ContainerCreating	0	5s
infinispn-operator-0	1/1	Running	0	3m
example-rhdg-3	1/1	Running	0	8s
example-rhdg-2	1/1	Running	0	8s
example-rhdg-1	1/1	Running	0	8s

## 2.2. CREATING DATA GRID CLUSTERS WITH CREDENTIALS

Create secrets that contain credentials so application users can authenticate to Data Grid nodes. You can then specify the name of the secret in a custom resource **yaml** and spin up clusters with the Data Grid Operator.

## Procedure

1. Create a secret that contains credentials for the application user.
  - a. Add an authentication secret **yaml**, for example:

```
$ cat > connect_secret.yaml<<EOF
apiVersion: v1
kind: Secret
metadata:
  name: connect-secret
type: Opaque
stringData:
  username: developer
  password: changeme
EOF
```

The secret must:

- Be **type: Opaque**.
- Have **username** and **password** fields.



### IMPORTANT

The preceding authentication secret contains sample values for the **username** and **password** fields. You should replace these values with credentials that conform to your organization's security requirements.

- b. Apply the authentication secret **yaml**.

```
$ oc apply -f connect_secret.yaml
```

2. Create a custom resource **yaml** for the cluster that includes the name of the secret that contains your credentials.

```
$ cat > cr_minimal_with_auth.yaml<<EOF
apiVersion: infinispn.org/v1
kind: Infinispn
metadata:
  name: example-rhdg
spec:
  replicas: 2
  connector:
    authentication:
      type: Credentials
      secretName: connect-secret
EOF
```

- **replicas** specifies the number of nodes in the Data Grid cluster.
- **connector** configures how users connect to Data Grid nodes to store and retrieve data.

3. Apply the custom resource **yaml**.

```
$ oc apply -f cr_minimal_with_auth.yaml
```

- Verify that the Data Grid Operator creates the pods.

```
$ oc get pods -w
```

NAME	READY	STATUS	RESTARTS	AGE
example-rhdg-1	0/1	ContainerCreating	0	4s
example-rhdg-2	0/1	ContainerCreating	0	4s
example-rhdg-3	0/1	ContainerCreating	0	5s
infinispan-operator-0	1/1	Running	0	3m
example-rhdg-3	1/1	Running	0	8s
example-rhdg-2	1/1	Running	0	8s
example-rhdg-1	1/1	Running	0	8s

## 2.3. VERIFYING DATA GRID CLUSTERS

Review log messages to verify that the Data Grid Operator has successfully created a Data Grid cluster.

### Procedure

- Verify that the Data Grid nodes can discover each other, for example:

```
$ oc logs example-rhdg-0 | grep DNS_PING
```

```
INFO Configuring JGroups discovery protocol to openshift.DNS_PING
INFO [org.jgroups.protocols.openshift.DNS_PING] (MSC service thread 1-2) serviceName
[my-namespace] set; clustering enabled
```

- Verify that the nodes have received a clustered view. Do either of the following:

- Retrieve the cluster view from the pod log files.

```
$ oc logs example-rhdg-0 | grep ISPN000094
```

```
INFO [org.infinispan.CLUSTER] (MSC service thread 1-2) \
ISPN000094: Received new cluster view for channel cluster: \
[example-rhdg-0|0] (1) [example-rhdg-0]
```

```
INFO [org.infinispan.CLUSTER] (jgroups-3,{example_crd_name-0}) \
ISPN000094: Received new cluster view for channel cluster: \
[example-rhdg-0|1] (2) [example-rhdg-0, example-rhdg-1]
```

- Retrieve the custom resource type for the Data Grid Operator.

```
$ oc get infinispan -o yaml
```

The output of the preceding command contains the following message to indicate that the Data Grid nodes have received a clustered view:

```
conditions:
  - message: 'View: [example-rhdg-0, example-rhdg-1]'
    status: "True"
```

█ type: wellFormed

## CHAPTER 3. MANAGING CLUSTER CREDENTIALS

Perform tasks to manage authentication for Data Grid clusters.

### Prerequisites

- An **oc** client in your **\$PATH**.

### 3.1. RETRIEVING CLUSTER CREDENTIALS

You can retrieve credentials for your Data Grid clusters as base64-encoded strings from secrets in your cluster namespace.

If you do not create secrets and credentials when you create clusters, the Data Grid Operator automatically generates them with the following defaults:

#### Default usernames

- Management user is **admin**.
- Application user is **developer**.

#### Default credentials secrets

- **example-rhdg-mgmt-generated-secret** contains credentials for the management user.
- **example-rhdg-app-generated-secret** contains credentials for the application user.

### Procedure

- Get the credentials from the secret. For example, to get the password for the application user from the default secret:

```
$ oc get secret example-rhdg-app-generated-secret \
-n my_namespace -o jsonpath="{.data.password}" | base64 --decode
```

#### TIP

Use the **jq** JSON processor to retrieve credentials as follows:

```
$ oc get secret example-rhdg-app-generated-secret \
-n my_namespace -o json | jq '.data | map_values(@base64d)'

{
  "password": "tUElqbfoJmT,NJVN",
  "username": "developer"
}
```