Redis Enterprise Software on Kubernetes deployment with the OpenShift OperatorHub

The OperatorHub is a feature of OpenShift 4.x that enables browsing a catalog of open-source and vendor-provided operators. You can install operators from the operator catalog in the OpenShift administrative interface with a few simple steps.

You can install the operator via the Command-Line Interface (CLI), but the OperatorHub is an easy way to help you to manage the operator and apply upgrades. It also provides an easy way to create clusters.

The operator installation process is:

1. Install the operator from the OperatorHub.
2. Use a Custom Resource Definition (CRD) to create clusters from the Operator view in the OperatorHub.
3. Create databases in Redis Enterprise.

Now the databases are exposed as Kubernetes services accessible by other application workloads in your K8s cluster.

Preparing the cluster

The Redis Enterprise node pods must run with certain privileges that are set in OpenShift using a Security Context Constraint that grants the pod various rights, such as the ability to change system limits or run as a particular user. At minimum, the security context constraint for the operator (scc.yaml) must be installed into the cluster as it is used by the OperatorHub installer. Without this constraint installed, the operator cannot create Redis Enterprise clusters.

The security context constraint for the operator needs to be installed only once and must not be deleted.

The constraint scc.yaml can be downloaded and installed by a cluster administrator with the commands:

```bash
curl -O https://raw.githubusercontent.com/RedisLabs/redis-enterprise-k8s-docs/master/openshift/scc.yaml
oc apply -f scc.yaml
```

After the constraint is installed, the OperatorHub automatically uses the constraint for Redis Enterprise node pods.

Note:

**Known Limitation** - The automatic use of the security constraint is limited. The Redis Enterprise must be named rec for the constraint to be used automatically. This limitation may be removed in the future. **We recommended that you use the cluster name rec when deploying with the OperatorHub.**

If you require a different name, you must grant the SCC to the project namespace (e.g., my-project) as in OpenShift 3.x:

```bash
oc adm policy add-scc-to-group redis-enterprise-scc system:serviceaccounts:my-project
```

Install the Operator

This procedure shows how you can install the Redis Enterprise Operator into a single project or namespace.

**Step 1: Log in**
The operator is deployed into a project. As such, you need an account with access to your project or the ability to create a new project.

1. Log in to your OpenShift account.

2. Either navigate to your project to set it as the default, or go to Projects and select Create Project.

In the dialog, enter the project name with the optional metadata and select Create.

Step 2: Navigate to database operators

1. In the left menu, go to Catalog > OperatorHub.

2. In the OperatorHub view, you see a list of categories on the left and a list of operators on the right. You can search for “Redis Enterprise” in the search dialog below the list of categories or select Database in the OperatorHub.

3. Make sure that your project is shown in the “Project:” label at the top of the page.

Step 3: Find and select install
1. Either search for “Redis Enterprise” or scroll through the various database operators and select the Redis Enterprise Operator.

![Redis Enterprise Operator](image)

Redis Enterprise Operator
provided by Redis Labs

Redis Enterprise Software is enterprise grade, distributed, in-memory NoSQL database server, fully compatible with open source.

2. Then select the Install button.

![Redis Enterprise Operator](image)

Redis Enterprise Operator
5.4.14-7 provided by Redis Labs

| Install | Redis Enterprise Software is compatible with open source, delivers stable high performance, operational savings. Redis Enterprise focus on rapid, high-quality development. With Redis Enterprise, you can:
| OPERATOR VERSION | 5.4.14-7 |
| PROVIDER TYPE | Certified |
| PROVIDER | Redis Labs |

- Enjoy high performance and stability
- Develop highly scalable and fault-tolerant systems
- Use versatile data structures and key-value storage

**Step 4: Create the operator subscription**

When you use the OperatorHub, a deployed operator is maintained by the OpenShift Operator Lifecycle Management. The Redis Enterprise subscription that provides this is only for a single project. You cannot select All namespaces on the cluster.

Note: We recommend that you change the Approval Strategy of the subscription from Automatic to Manual for production systems so that the operator is only upgraded by approval. Then you can upgrade during maintenance periods to limit downtime due to upgrades to operator-managed clusters.

- Click Subscribe to start the subscription for your project.
The subscription is shown and the installation status is updated soon after. If you selected Manual approvals, the requests approval link is shown.
Step 5: Preview and approve install plan

1. Click requires approval to see the install plan that requires approval.

2. Click Preview Install Plan to preview the install plan.

redis-enterprise-operator-cert > Install Plan Details

install-87815

3. Select the Approve button to install the operator.
redis-enterprise-operator.v5.4.14-7

NAME

CSV  redis-enterprise-operator.v5.4.14-7
CRD  redisenterpriseclusters.app.redislabs.com

After the approval, the status of the resources installed is updated.

**Step 6: Return to the operator**

1. Click `redis-enterprise-operator-cert` to select the operator.
   
   **Project: my-project**

2. Click on the installed version to see more information about the current version and create clusters.

3. Click on the catalog source to see the subscription.

The installed version and catalog source are shown along with other information.

- Click on the installed version for information about the current version and create clusters.
- Click on the catalog source to see the subscription.

2. Click on the installed version to see more information about the current version and create clusters.

3. Click on The catalog source to view the subscription.

**Next steps**

After you install the Redis Enterprise operator into a project you can:
Navigate to the installed version and use the Create New button to create a new cluster in your namespace.

Click on Catalog > Installed Operators in the left menu to view the operators in your project.

Click on Catalog > Operator Management to view your operator subscriptions.

Create a Redis Enterprise cluster

You can easily create a cluster with the operator in Installed Operators.

Step 1: Navigate to the installed operator

1. Click on Catalog > Installed Operators.

2. Find the Redis Enterprise Operator listing for your project.

Step 2: Access the operator API

1. Under Provided APIs for that listing, select RedisEnterpriseCluster.

2. Select Create Redis Enterprise Cluster to create a cluster.

Redis Enterprise Clusters

Create Redis Enterprise Cluster

The CRD for a Redis Enterprise cluster in YAML format is shown.

Step 3: Configure and create your cluster
1. Verify that you are creating the CRD in the correct project namespace. You can see this above the text edit and shown in the YAML located after namespace field:

![Project: my-project ▼](image)

Create Redis Enterprise Cluster
Create by manually entering YAML or JSON

```yaml
apiVersion: app.redislabs.com/v1
kind: RedisEnterpriseCluster
metadata:
  name: rec
  namespace: my-project
spec:
```

2. Edit the custom resource definition.

![Note: The name of the cluster must be "rec" for deployments from within the OLM. For more information, see the name: field description in Options for clusters.](image)

3. Click Create to create and deploy the cluster.

<table>
<thead>
<tr>
<th>NAME</th>
<th>LABELS</th>
<th>TYPE</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>rec</td>
<td>No labels</td>
<td>RedisEnterpriseCluster</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

The name of your cluster is shown in the list and you click on the cluster name to see more details about the cluster.

Create a database

**Step 1: Access the Redis Enterprise UI**

When the cluster is created, a login is generated using the username specified in the cluster CRD (e.g., demo@redislabs.com) and a password is generated by the operator. The generated password is stored in a Kubernetes secret.

1. Go to Workloads > Secrets in the left menu:
2. Find the cluster name used in the CRD and click on the name to see your secret.

3. Scroll to the Data section to find the username and password.

   Data

   LICENSE
   No value

   PASSWORD
   · · · ·

   USERNAME
   · · · ·

4. Click copy to clipboard or Reveal Values to save the login information for later use.

5. Go to Workloads > Pods to see the pods associated with the cluster. These all have a number starting at “0” (e.g., rec-0) at the end of the name. Each of these pods has an instance of the Redis Enterprise UI running in it on port 8843.

6. To forward the port of one of these pods to your local machine, run:

   kubectl port-forward rec-0 8443:8443

   Where rec-0 is replaced with of the pods associated with your Redis Enterprise cluster.

   Note:
   - rec-0 is one of your cluster pods. Consider running the port-forward command in the background.
   - The Openshift UI provides tools for creating additional routing options, including external routes.
7. In a browser, go to localhost:8443 to open the Redis Enterprise admin console:

8. Enter the username and password to log in to your cluster.

Step 2: Create your database

Continue with the instructions to create your database. After you create a database, the operator discovers the database and creates additional Kubernetes services for application workload access to the database.

Step 3: Inspect your database services
After you create your database in the Redis Enterprise admin console, the operator detects the change and creates Kubernetes services that expose the database. The databases are named according to the database name. For example, if you called your database "test", kubectl shows these services:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>PORT(S)</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>rec</td>
<td>ClusterIP</td>
<td>None</td>
<td>&lt;none&gt;</td>
<td>9443/TCP, 8001/TCP, 8070/TCP</td>
<td>43m</td>
</tr>
<tr>
<td>rec-ui</td>
<td>ClusterIP</td>
<td>172.30.16.113</td>
<td>&lt;none&gt;</td>
<td>8443/TCP</td>
<td>43m</td>
</tr>
<tr>
<td>test</td>
<td>ClusterIP</td>
<td>172.30.212.143</td>
<td>&lt;none&gt;</td>
<td>13818/TCP</td>
<td>20s</td>
</tr>
<tr>
<td>test-headless</td>
<td>ClusterIP</td>
<td>None</td>
<td>&lt;none&gt;</td>
<td>13818/TCP</td>
<td>20s</td>
</tr>
</tbody>
</table>

The operator creates two services, shown here as "test" and "test-headless". The service with the "-headless" suffix is a Kubernetes headless service and does not provide load balancing.

Updated: March 23, 2022