Delete custom resources

Delete a database (REDB)
To delete a database managed by the Redis Enterprise Kubernetes operator, run `kubectl delete redb <your-db-name>` from your K8s cluster.

When you delete a database, your data and the REDB custom resource are also deleted.

Delete a Redis Enterprise cluster (REC)
To delete a Redis Enterprise cluster managed by the operator:

1. Delete all the databases in your cluster.
2. Run `kubectl delete rec <your-rec-name>` from your K8s cluster.

When you delete your cluster, your databases and the REC custom resource are also deleted. However, persistent volume claims (PVCs) for your cluster are not deleted in the process. If you want to delete your PVCs, you’ll have to delete them manually.

Delete the operator
To delete the operator from your K8s cluster and namespace, you can delete the operator bundle with:

- `kubectl delete -f bundle.yaml` for non-OpenShift K8s deployments
- `kubectl delete -f openshift.bundle.yaml` for OpenShift deployments

This will remove the operator and its custom resource definitions (CRDs) from your K8s cluster.

**Warning** - The Redis Enterprise CRDs are non-namespaced resources, meaning they are shared across your entire K8s cluster. Deleting CRDs in one namespace will delete custom resources in every other namespace across the K8s cluster.

If you have Redis Enterprise clusters running in different namespaces on the same K8s cluster, deleting the entire operator bundle might cause data loss.

To safely delete the operator from one namespace without affecting the others, delete the operator files individually, excluding the CRD files:

```
kubectl delete -f role.yaml
kubectl delete -f role_binding.yaml
kubectl delete -f service_account.yaml
kubectl delete -f admission-service.yaml
kubectl delete -f operator.yaml
```

You will also need to remove the `namespaceSelector` section from the validating webhook.

Troubleshoot REDB deletion
The operator attaches a finalizer to the Redis Enterprise database (REDB) object. This makes sure the database is deleted before the REDB custom resource is removed from the K8s cluster.

If the operator isn’t running, or some other fatal error occurs, the finalizer isn’t removed automatically by the operator. In this case, you won’t be able to delete your REDB resource.
If this happens, you can remove the finalizer manually.

```
kubectl patch redb <your-db-name> --type=json -p '{"op":"remove","path":"/metadata/finalizers","value":"finalizer.redisenterprisedatabases.app.redislabs.com"}'
```

**Troubleshoot REC deletion**

The operator attaches a finalizer to the Redis Enterprise cluster (REC) object. This makes sure the Redis cluster is deleted before the REC custom resource is removed from the K8s cluster.

If the operator isn’t running, or some other fatal error occurs, the finalizer isn’t removed automatically by the operator. In this case, you won’t be able to delete your REDB resource.

If this happens, you can remove the finalizer manually.

```
kubectl patch rec <your-rec-name> --type=json -p '{"op":"remove","path":"/metadata/finalizers","value":"redbfinalizer.redisenterpriseclusters.app.redislabs.com"}'
```

*Updated: April 6, 2022*