Recover a Redis Enterprise cluster on Kubernetes

When a Redis Enterprise cluster loses contact with more than half of its nodes either because of failed nodes or network split, the cluster stops responding to client connections. When this happens, you must recover the cluster to restore the connections.

You can also perform cluster recovery to reset cluster nodes, to troubleshoot issues, or in a case of active/passive failover.

The cluster recovery for Kubernetes automates these recovery steps:

1. Recreates a fresh Redis Enterprise cluster
2. Mounts the persistent storage with the recovery files from the original cluster to the nodes of the new cluster
3. Recovers the cluster configuration on the first node in the new cluster
4. Joins the remaining nodes to the new cluster.

Prerequisites

- For cluster recovery, the cluster must be deployed with persistence.
- For data recovery, the databases must be configured with persistence.

Recovering a cluster on Kubernetes

To recover a cluster on Kubernetes:

1. Edit the rec resource to set the clusterRecovery flag to true, run:

   ```bash
   kubectl patch rec <cluster-name> --type merge --patch '{"spec":
   {"clusterRecovery":true}}'
   ```

   **Note:**
   In some cases, pods do not terminate when the statefulSet is scaled down as part of the cluster recovery. If pods are stuck in terminating or crashLoopBack and do not terminate gracefully, cluster recovery can pause. To work around this, delete the pods manually with:

   ```bash
   kubectl delete pods <pod> --grace-period=0 --force
   ```

   When the last pod is manually deleted, the recovery process resumes.

2. Wait for the cluster to recover until it is in the Running state.

   To see the state of the cluster, run:

   ```bash
   watch "kubectl describe rec | grep State"
   ```

3. To recover the cluster data, once the cluster is in Running state, for any cluster pod run:
kubectl exec <pod-name> -- radmin recover all

This command recovers the data for all nodes in the cluster based on the cluster configuration in pod-0.

Note: If the database status is missing files, make sure all persistence files are placed on the correct nodes under the persistence path. In case of databases with AOF persistence enabled, you may need to rename AOF files on the pods to remove the .prev suffix.

If you want to recover based on the cluster configuration of another pod, copy the cluster configuration from the source pod (/var/opt/redislabs/persist/ccb/ccb-redis.rdb) to pod-0.

4. If you are using sentinel discovery service, you must restart the sentinel_service on the master. To do this, log into the master pod and run:

    supervisorctl restart sentinel_service

Updated: April 14, 2022