Maintenance mode for cluster nodes

Use maintenance mode to prevent data loss during hardware or operating system maintenance on Redis Enterprise servers. When maintenance mode is on, all shards move off of the node under maintenance and migrate to another available node.

Turn maintenance mode on

When you turn maintenance mode on, Redis Enterprise does the following:

1. Checks whether a shut down of the node will cause quorum loss. If so, maintenance mode will not turn on.

   Warning - Maintenance mode does not protect against quorum loss. If you turn on maintenance mode for the majority of nodes in a cluster and restart them simultaneously, the quorum is lost. This can cause data loss.

2. Takes a snapshot of the node configuration to record which shards and endpoints are on the node.

3. Marks the node as a quorum node to prevent shards and endpoints from migrating to it.

   Note: If you run rladmin status, the node's shards field is now yellow to show that shards cannot migrate to it.

4. Migrates shards and binds endpoints to other nodes, if space is available.

   Note: Maintenance mode does not demote a master node by default. The cluster elects a new master node when the original master node is restarted. Alternatively, you can add the demote_node option to the rladmin command to demote a master node when you turn on maintenance mode.

To turn on maintenance mode for a node, run the following command:

```
rladmin node <node_id> maintenance_mode on
```

You can start server maintenance if:

- All shards and endpoints have moved to other nodes
- Enough nodes are still online to maintain quorum

Prevent replica shard migration

If you do not have enough resources available to move all of the shards to other nodes, you can turn maintenance mode on without migrating the replica shards.
To turn on maintenance mode without replica shard migration, run:

```
rladmin node <node_id> maintenance_mode on keep_slave_shards
```

**Demote a master node**

If you need to do maintenance that might affect connectivity to the master node, you can demote the master node when you turn on maintenance mode. This process also allows the cluster to elect a new master quicker.

To demote a master node when you turn on maintenance mode, run:

```
rladmin node <node_id> maintenance_mode on demote_node
```

**Turn maintenance mode off**

When you turn maintenance mode off, Redis Enterprise:

1. Loads a specified snapshot or defaults to the latest snapshot.
2. Unmarks the node as a quorum node to allow shards and endpoints to migrate to the node.
3. Restores the shards and endpoints that were in the node at the time of the snapshot.
4. Deletes the snapshot.

To turn maintenance mode off after you finish server maintenance, run:

```
rladmin node <node_id> maintenance_mode off
```

**Specify a snapshot**

Redis Enterprise saves a snapshot of the node configuration every time you turn on maintenance mode. If multiple snapshots exist, you can restore a specific snapshot when you turn maintenance mode off.

To get a list of available snapshots, run:

```
rladmin node <node_id> snapshot list
```

To specify a snapshot when you turn maintenance mode off, run:

```
rladmin node <node_id> maintenance_mode off snapshot_name <snapshot_name>
```

**Note:** If an error occurs when you turn on maintenance mode, the snapshot is not deleted. When you rerun the command, use the snapshot from the initial attempt since it contains the original state of the node.

**Skip shard restoration**

You can prevent the migrated shards and endpoints from returning to the original node after you turn off maintenance mode.

To turn maintenance mode off and skip shard restoration, run:
Cluster status example

This example shows how the output of `rladmin status` changes when you turn on maintenance mode for a node.

The cluster status before turning on maintenance mode:

```bash
redislabs@rp1_node1:/opt$ rladmin status
CLUSTER NODES:
NODE:ID ROLE ADDRESS EXTERNAL_ADDRESS HOSTNAME SHARDS
*node:1 master 172.17.0.2 rp1_node1 2/100
node:2 slave 172.17.0.4 rp3_node1 2/100
node:3 slave 172.17.0.3 rp2_node1 0/100
```

The cluster status after turning on maintenance mode:

```bash
redislabs@rp1_node1:/opt$ rladmin node 2 maintenance_mode on
Performing maintenance_on action on node:2: 0%
created snapshot NodeSnapshot<name=maintenance_mode_2019-03-14_09-50-59,time=None,node_uid=2>
node:2 will not accept any more shards
Performing maintenance_on action on node:2: 100%
OK
redislabs@rp1_node1:/opt$ rladmin status
CLUSTER NODES:
NODE:ID ROLE ADDRESS EXTERNAL_ADDRESS HOSTNAME SHARDS
*node:1 master 172.17.0.2 rp1_node1 2/100
node:2 slave 172.17.0.4 rp3_node1 0/0
node:3 slave 172.17.0.3 rp2_node1 2/100
```

After turning on maintenance mode for node 2, Redis Enterprise saves a snapshot of its configuration and then moves its shards and endpoints to node 3.

Now node 2 has 0/0 shards because shards cannot migrate to it while it is in maintenance mode.

Toggle maintenance mode via API

You can also turn maintenance mode on or off via REST API requests to `POST /nodes/{node_uid}/actions/{action}`.

Maintenance mode on

Send a POST `/nodes/{node_uid}/actions/maintenance_on` request to turn on maintenance mode:

```bash
curl -X POST https://<hostname>:9443/v1/nodes/<node_id>/actions/maintenance_on
-k -u <user>:<password>
--data '{"keep_slave_shards":true}'
-H "Content-Type: application/json"
```

The `keep_slave_shards` boolean flag prevents replica shard migration when set to true.

The maintenance_on request returns a JSON response body:
**Maintenance mode off**

Send a `POST /nodes/{node_uid}/actions/maintenance_off` request to turn off maintenance mode:

```bash
curl -X POST https://<hostname>:9443/v1/nodes/<node_id>/actions/maintenance_off
-k -u <user>:<password>
--data '{"skip_shards_restore":false}'
-H "Content-Type: application/json"
```

The `skip_shards_restore` boolean flag allows the `maintenance_off` action to **skip shard restoration** when set to `true`.

The `maintenance_off` request returns a JSON response body:

```json
{
  "status": "queued",
  "task_id": "38c7405b-26a7-4379-b84c-cab4b3db706d"
}
```

**Track action status**

You can send a request to `GET /nodes/{node_uid}/actions/{action}` to track the `status` of the `maintenance_on` and `maintenance_off` actions.

This request returns the status of the `maintenance_on` action:

```bash
curl https://<hostname>:9443/v1/nodes/<node_id>/actions/maintenance_on
-k -u <user>:<password>
```

The response body:

```json
{
  "status": "completed",
  "task_id": "38c7405b-26a7-4379-b84c-cab4b3db706d"
}
```

**Updated**: November 29, 2021