Create and manage cloud accounts

You can use the Redis Enterprise Cloud REST API to create and manage cloud accounts.

These examples use the cURL utility; you can use any REST client to work with the Redis Cloud REST API.

Create a cloud account

To create a cloud account, use:

```
POST /cloud-accounts
```

The following Linux shell script sends a POST `/cloud-accounts` and waits for a cloud account ID. When the cloud account ID is received, the processing phase runs.

Prerequisites

Before you use the API to create and manage a cloud account, you must:

- These examples require `jq`, a JSON parser.
  - Use your package manager to install it (Example: `sudo apt install jq`)
- Define the expected variables needed to use the API:

  ```
  export HOST=api.redislabs.com/v1
  export ACCOUNT_KEY={replace-with-your-account-key}
  export SECRET_KEY={replace-with-your-secret-key}

  export ACCESS_KEY_ID={replace-with-your-access-key-id}
  export ACCESS_SECRET_KEY={replace-with-your-access-secret-key}
  export CONSOLE_PASSWORD={replace-with-your-console-password}
  export CONSOLE_USERNAME={replace-with-your-console-username}
  export AWS_ACCOUNT_IDENTIFIER={replace-with-your-aws-account-identifier}
  ```

Cloud account JSON body

The created cloud account is defined by a JSON document that is sent as the body of the POST `/cloud-accounts` request.

In the example below, that JSON document is stored in the `create-cloud-account-basic.json` file:

```json
{
  "accessKeyId": "$ACCESS_KEY_ID",
  "accessSecretKey": "$ACCESS_SECRET_KEY",
  "consolePassword": "$CONSOLE_PASSWORD",
  "consoleUsername": "$CONSOLE_USERNAME",
  "name": "My new Cloud Account",
  "provider": "AWS",
  "signInLoginUrl": "https://$AWS_ACCOUNT_IDENTIFIER.signin.aws.amazon.com/console"
}
```
Cloud account creation script

You can run the **create cloud account** script from the command line with:
```
bash path/script-name.sh
```

Below is the sample script that you can use as a reference to call the API operation to create a cloud account. The script contains the steps that are explained below.

```bash
echo "Step 1: Post a request to create a cloud account"

TASK_ID=$(curl -s -X POST "https://$HOST/cloud-accounts" \
         --header "Content-Type: application/json" \
         -H "accept: application/json" \
         -H "x-api-key: $ACCOUNT_KEY" \
         -H "x-api-secret-key: $SECRET_KEY" \
         --data-binary "@./static/code/rv/api/create-cloud-account-basic.json" \
         | jq -r .taskId)

echo "TASK_ID=$TASK_ID"

echo "Step 2: wait for processing completion"

STATUS=
while [ "$STATUS" != 'processing-completed' ] || [ "$STATUS" != 'processing-error' ]
do
    sleep 3 # seconds
    STATUS=$(curl -s -X GET "https://$HOST/tasks/$TASK_ID" \
            -H "accept: application/json" \
            -H "x-api-key: $ACCOUNT_KEY" \
            -H "x-api-secret-key: $SECRET_KEY" \
            | jq -r .status)
    echo "STATUS=$STATUS"
don

echo "Step 3: print the response, containing the create resource Id"

echo "Response: "
curl -s -X GET "https://$HOST/tasks/$TASK_ID" \
      -H "accept: application/json" \
      -H "x-api-key: $ACCOUNT_KEY" \
      -H "x-api-secret-key: $SECRET_KEY" \
      | jq -r .
```

**Step 1 of the cloud account creation script**

This step executes a POST request to `cloud-accounts` with the defined parameters and a JSON body located within a separate JSON file.

The POST response is a JSON document that contains the `taskId`, which is stored in a variable called `TASK_ID` that is used later to track the progress of the request.
Step 2 of the cloud account creation script

This step queries the API for the status of the cloud account creation request based on the taskId stored in the $TASK_IS variable.

Step 3 of the cloud account creation script

When the status changes from processing-in-progress to processing-completed (or processing-error), this step prints the response, including the resourceId. In this case, the resourceId is a cloud account ID.

If the processing phase completed successfully, the cloud account is displayed in the admin console with a pending status. This indicates that the cloud account is being provisioned.

You can use the GET /cloud-accounts/{cloudAccountId} API operation to track the created cloud account until it becomes active.

Updated: August 10, 2021