Database backup and import

Back up a database

When you create or update a database in a Flexible or Annual account, you can specify the optional `periodicBackupPath` parameter with a backup path. This parameter enables periodic and on-demand backup operations for the specified database.

The API operation for on-demand backups is POST `/subscriptions/{subscriptionId}/databases/{databaseId}/backup`. On-demand database backup is an asynchronous operation.

Prerequisites for backups

Before you enable backups, you must define the variables that the API requires:

```bash
export HOST=api.redislabs.com/v1
export ACCOUNT_KEY={replace-with-your-account-key}
export SECRET_KEY={replace-with-your-secret-key}
export SUBSCRIPTION_ID={subscription-id}
export DATABASE_ID={database-id}
```

Database backup script

```bash
curl -s -X POST 'https://$HOST/subscriptions/$SUBSCRIPTION_ID/databases/$DATABASE_ID/backup' \
    --header 'Content-Type: application/json' \
    -H 'accept: application/json' \
    -H 'x-api-key: $ACCOUNT_KEY' \
    -H 'x-api-secret-key: $SECRET_KEY'
```

The backup database API does not require a body. Instead, the `periodicBackupPath` must be set to a valid path with available storage capacity to store the backup files for the specific database.

Import a database

You can import data into an existing database from multiple storage sources, including AWS S3, Redis, and FTP. Database import is an asynchronous operation.

The API operation for performing on-demand backup is POST `/subscriptions/{subscriptionId}/databases/{databaseId}/import`.

The requirements for data import are:

- The URI of the data
- The source URI must be accessible to the importing database
- The data format must be a Redis backup file or a Redis database
- The subscription ID and database ID of the destination database

The duration of the import process depends on the amount of data imported and the network bandwidth between the data source and the importing database.

Warning - Data imported into an existing database overwrites any existing data.

Database import script

Before you import the data, you must set the import variables:

```bash
export HOST=api.redislabs.com/v1
export ACCOUNT_KEY={replace-with-your-account-key}
export SECRET_KEY={replace-with-your-secret-key}
export SUBSCRIPTION_ID={subscription-id}
export DATABASE_ID={database-id}
```

To import the data, run:
```
curl -s -X POST "https://$HOST/subscriptions/$SUBSCRIPTION_ID/databases/$DATABASE_ID/import" \
   --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/import-database-s3.json"
```

**Database import JSON body**

The database import operation is defined by a JSON document that is sent as the body of the API request.

In the example above, that JSON document is stored in the `import-database-s3.json` file:

```json
{
   "sourceType": "aws-s3",
   "importFromUri": [
      "s3://bucketname/filename-dbForAWSBackup-1_of_3.rdb.gz",
      "s3://bucketname/filename-dbForAWSBackup-2_of_3.rdb.gz",
      "s3://bucketname/filename-dbForAWSBackup-3_of_3.rdb.gz"
   ]
}
```

You can specify the backup location with the `sourceType` and `importFromUri` values for these sources:

<table>
<thead>
<tr>
<th>Data location</th>
<th>sourceType</th>
<th>importFromUri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon AWS S3</td>
<td>aws-s3</td>
<td>s3://bucketname/[path/][filename.rdb[.gz]</td>
</tr>
<tr>
<td>FTP</td>
<td>ftp</td>
<td>ftp://[username][password]@[port]/[path/][filename.rdb[.gz]</td>
</tr>
<tr>
<td>Google Blob Storage</td>
<td>google-blob-storage</td>
<td>gs://bucketname/[path/][filename.rdb[.gz]</td>
</tr>
<tr>
<td>Microsoft Azure Blob Storage</td>
<td>azure-blob-storage</td>
<td>abs://storage_account_access_key@storage_account_name[container/][filename.rdb[.gz]</td>
</tr>
<tr>
<td>Redis server</td>
<td>redis</td>
<td>redis:[db_password]@[host]:[port]</td>
</tr>
<tr>
<td>Web server</td>
<td>HTTP</td>
<td>HTTP://[username][password]@[port]/[path/][filename.rdb[.gz]</td>
</tr>
</tbody>
</table>

Updated: February 2, 2022