Export data from a database

You can export the data from a specific database at any time. The following destinations are supported:

- FTP server
- SFTP server
- Amazon AWS S3
- Local mount point
- Azure Blob Storage
- Google Cloud Storage

If you export a database configured for database clustering, export files are created for each shard.

Storage space requirements

Before exporting data, verify that you have enough space available in the storage destination and on the local storage associated with the node hosting the database.

Export is a two-step process: a temporary copy of the data is saved to the local storage of the node and then copied to the storage destination. (The temporary file is removed after the copy operation.)

Export fails when there isn't enough space for either step.

Export database data

To export data from a database:

1. Sign in to the admin console.
2. Select the Databases command from the console menu.
3. Select the database from the database list.
4. Select the Configuration tab.
5. Select the Export button.

If the Export button is disabled, you do not have permission to export data.
6. Enter the export details.

Choose storage type

FTP

Path

Receive email notification on success / failure

Choose storage type list defines the destination storage container for the exported data; select the appropriate value and then enter the requested details. Details vary for each storage type. For help, see Supported storage locations.

7. Select the Export button to begin the export process.

Supported storage locations

Data can be exported to a local mount point, transferred to a URI using FTP/SFTP, or stored on cloud provider storage.

When saved to a local mount point or a cloud provider, export locations need to be available to the group and user running Redis Enterprise Software, redislabs:redislabs by default.

Redis Enterprise Software needs the ability to view permissions and update objects in the storage location. Implementation details vary according to the provider and your configuration. To learn more, consult the provider’s documentation.

The following sections provide general guidelines. Because provider features change frequently, use your provider’s documentation for the latest info.

FTP server

Before exporting data to an FTP server, verify that:

- Your Redis Enterprise cluster can connect and authenticate to the FTP server.
- The user specified in the FTP server location has read and write privileges.

To export data to an FTP server, set Path using the following syntax:

ftp://[username]:[password]@[host]:[port]/[path]/

Where:
- **protocol**: the server's protocol, can be either ftp or ftps.
- **username**: your username, if needed.
- **password**: your password, if needed.
- **hostname**: the hostname or IP address of the server.
- **port**: the port number of the server, if needed.
- **path**: the export destination path, if needed.

Example: `ftp://username:password@10.1.1.1/home/exports/`

The user account needs permission to write files to the server.

### SFTP server

Before exporting data to an SFTP server, make sure that:

- Your Redis Enterprise cluster can connect and authenticate to the SFTP server.
- The user specified in the SFTP server location has read and write privileges.
- The SSH private keys are specified correctly. You can use the key generated by the cluster or specify a custom key.

When using the cluster key, copy the **Cluster SSH Public Key** to the appropriate location on the SFTP server. This is available from the **General** tab of the **Settings** menu in the admin console.

Use the server documentation to determine the appropriate location for the SSH Public Key.

To export data to an SFTP server, enter the SFTP server location in the format:

```
sftp://user:password@host:<custom_port>/path/
```

For example: `sftp://username:password@10.1.1.1/home/exports/`

### Local mount point

Before exporting data to a local mount point, verify that:

- The node can connect to the destination server, the one hosting the mount point.
- The **redislabs**: **redislabs** user has read and write privileges on the local mount point and on the destination server.
- The export location has enough disk space for your exported data.

To export to a local mount point:

1. On each node in the cluster, create the mount point:
   1. Connect to a shell running Redis Enterprise Software server hosting the node.
   2. Mount the remote storage to a local mount point.

For example:
sudo mount -t nfs 192.168.10.204:/DataVolume/Public /mnt/Public

2. In the path for the export location, enter the mount point.
   For example: /mnt/Public

3. Verify that the user running Redis Enterprise Software has permissions to access and update files in the mount location.

**AWS Simple Storage Service**

To export data to an Amazon Web Services (AWS) Simple Storage Service (S3) bucket:

1. Sign in to the AWS Management Console.
2. Use the Services menu to locate and select Storage > S3. This takes you to the Amazon S3 admin panel.
3. If you do not already have a bucket for exports, select the Create Bucket button in the upper, right corner of the Buckets panel.
   - When the Create bucket screen appears, enter a name for your bucket.
   - Set AWS Region to an appropriate region.
   - Set other properties according to your company standards.
   - When finished, select the Create bucket button near the bottom of the screen.
4. Use the Buckets list to locate and select your bucket. When the settings appear, select the Permissions tab, locate the Access control list (ACL) section, and then select the Edit button.
5. When the Edit access control list (ACL) screen appears, locate the Access for other AWS accounts section and then select the Add grantee button.
   - In the Grantee field, enter the AWS account ID: fd1b05415aa5ea3a310265ddb13b156c7c76260dbc87e037a8fc290c3c86b614
   - In the Objects list, enable Write.
   - In the Bucket ACL list, enable Read and Write.
   - When finished, select the Save changes button.

Once the bucket is available and the permissions are set, use the name of your bucket as the Backup destination for your database Remote backup settings.

Use the S3 protocol scheme (s3://) to set bucket-name to the name of your export bucket. If, for example, your bucket is named exports-bucket, set Path to s3://exports-bucket.

**GCP Storage**

For Google Cloud Platform (GCP) console subscriptions, store your exports in a Google Cloud Storage bucket:

2. In the admin console menu, locate the Storage section then select Cloud Storage > Browser.
3. Create or select a bucket.
4. Select the overflow menu (three dots, stacked) and then select the Edit Bucket Permissions command.
5. Select the Add members button and then add:

7. Save your changes.

8. Verify that your bucket does not have a set retention policy.

   To do so:
   1. View the details of your bucket.
   2. Select the Retention tab.
   3. Verify that there is no retention policy.

   If a policy is defined and you cannot delete it, you need to use a different bucket.

Use the bucket details Configuration tab to locate the gsutil URI. This is the value you’ll assign to your resource’s path.

**Azure Blob Storage**

To export to Microsoft Azure Blob Storage, sign in to the Azure portal and then:

1. Create an Azure Storage account if you do not already have one
2. Create a container if you do not already have one
3. Manage storage account access keys

Set your resource’s Path to the path of your storage account.

The syntax for creating the export destination varies according to your authorization mechanism. For example:

abs://storage_account_access_key@storage_account_name/container_name/[path/]

Where:

- **storage_account_access_key**: the primary access key to the storage account
- **storage_account_name**: the storage account name
- **container_name**: the name of the container, if needed.
- **path**: the backups path, if needed.

To learn more, see Authorizing access to data in Azure Storage.

**Updated:** October 31, 2021