Schedule periodic backups

Periodic backups provide a way to restore data with minimal data loss. With Redis Enterprise Software, you can schedule periodic backups to occur every once a day (every 24 hours), twice a day (every twelve hours), every four hours, or every hour.

As of v6.2.8, you can specify the start time for twenty-four or twelve hour backups.

To make an on-demand backup, export your data.

You can schedule backups to a variety of locations, including:

- FTP server
- SFTP server
- Local mount point
- Amazon Simple Storage Service (S3)
- Azure Blob Storage
- Google Cloud Storage

The backup process creates compressed (.gz) RDB files that you can import into a database.

When you back up a database configured for database clustering, Redis Enterprise Software creates a backup file for each shard in the configuration. All backup files are copied to the storage location.

Note:

- Make sure that you have enough space available in your storage location. If there is not enough space in the backup location, the backup fails.
- The backup configuration only applies to the node it is configured on.

Schedule periodic backups

Before scheduling periodic backups, verify that your storage location exists and is available to the user running Redis Enterprise Software (redislabs by default). You should verify that:

- Permissions are set correctly
- The user running Redis Enterprise Software is authorized to access the storage location
- The authorization credentials work

Storage location access is verified before periodic backups are scheduled.

To schedule periodic backups for a database:

1. Sign in to the Redis Enterprise Software admin console using admin credentials.
2. From the admin console, choose Databases and then select your database.
3. Select the Edit button.
4. Locate and enable the Periodic backup checkbox.
5. Use the following table to help specify the details:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval</td>
<td>Specifies the frequency of the backup; that is, the time between each backup snapshot. Supported values include Every 24 hours, Every 12 hours, Every 4 hours, and Every hour. v6.2.8 or later: Specifies the start time for the backup; available when Interval is set to Every 24 hours or Every 12 hours.</td>
</tr>
<tr>
<td>Set starting time</td>
<td>If not specified, defaults to a time selected by Redis Enterprise Software. Specifies the storage type for the backup. Supported options vary and might require additional details. To learn more, see Supported storage locations.</td>
</tr>
<tr>
<td>Choose storage type</td>
<td>Access to the storage location is verified when you apply your updates. This means the location, credentials, and other details must exist and function before you can enable periodic backups.</td>
</tr>
</tbody>
</table>

6. Select Update to apply your changes.

Default backup start time

If you do not specify a start time for twenty-four or twelve hour backups, Redis Enterprise Software chooses one for you, based on the time the backups are enabled.

This choice assumes that your database is deployed to a multi-tenant cluster containing multiple databases. This means that default start times are staggered (offset) to ensure availability. This is done by calculating a random offset which specifies a number of seconds added to the start time.

Here’s how it works:

- Assume you’re enabling the backup at 4:00 pm (1600 hours).
- You choose to back up your database every 12 hours.
- Because you didn’t set a start time, the cluster randomly chooses an offset of 4,320 seconds (or 72 minutes).

This means your first periodic backup occurs 72 minutes after the time you enabled periodic backups (4:00 pm + 72 minutes). Backups repeat every twelve hours at roughly same time.
The backup time is imprecise because they’re started by a trigger process that runs every five minutes. When the process wakes, it compares the current time to the scheduled backup time. If that time has passed, it triggers a backup.

If the previous backup fails, the trigger process retries the backup until it succeeds.

In addition, throttling and resource limits also affect backup times.

For help with specific backup issues, contact support.

**Supported storage locations**

Database backups can be saved 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, backup 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 enabling backups 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 store your backups on an FTP server, set its **Backup 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 backup path, if needed.

Example: ftp://username:password@10.1.1.1/home/backups/

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

**SFTP server**

Before enabling backups 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 auto generated 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 backup 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/backups/`

**Local mount point**

Before enabling periodic backups 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 backup location has enough disk space for your backup files. Backup files are saved with filenames that include the timestamp, which means that earlier backups are not overwritten.

To back up to a local mount point:

1. On each node in the cluster, create the mount point:
   1. Connect to a shell running on Redis Enterprise Software server hosting the node.
   2. Mount the remote storage to a local mount point.
      
      ```
      sudo mount -t nfs 192.168.10.204:/DataVolume/Public /mnt/Public
      ```
   
   2. In the path for the backup 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 store backups in 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 backups, select the Create Bucket button in the upper, right corner of the Buckets panel.
   1. When the Create bucket screen appears, enter a name for your bucket.
   2. Set AWS Region to an appropriate region.
   3. Set other properties according to your company standards.
   4. 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.
1. In the **Grantee** field, enter the AWS account ID:
   fd1b85415aa5ea3a310265dddb13b156c7c76260dbc87e037a8fc290c3c86b614

2. In the **Objects** list, enable **Write**.
3. In the **Bucket ACL** list, enable **Read** and **Write**.
4. 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 `backups-bucket` to the name of your backup bucket. If, for example, your bucket is named `backups-bucket`, set **Backup destination** to `s3://backups-bucket`.

### GCP Storage

For **Google Cloud Platform (GCP) console** subscriptions, store your backups 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:
   
   `service@redislabs-prod-clusters.iam.gserviceaccount.com`

6. Set **Role** to **Storage Legacy | Storage Legacy Bucket Writer**.
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 backup path.

### Azure Blob Storage

To store your backup in 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 **Backup Path** to the path of your storage account.

The syntax for creating the backup 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