



# Red Hat OpenStack Platform 17.1

## Backing up Block Storage volumes

Deploy and use the Red Hat OpenStack Platform Block Storage (cinder) backup service



# Red Hat OpenStack Platform 17.1 Backing up Block Storage volumes

---

Deploy and use the Red Hat OpenStack Platform Block Storage (cinder) backup service

OpenStack Team  
rhos-docs@redhat.com

## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Deploy and use the Red Hat OpenStack Platform Block Storage backup service to backup and restore Block Storage volumes.

## Table of Contents

|  |           |
|--|-----------|
| <b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....                           | <b>3</b>  |
| <b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....                 | <b>4</b>  |
| <b>CHAPTER 1. BLOCK STORAGE BACKUP SERVICE OVERVIEW</b> .....            | <b>5</b>  |
| 1.1. BACKUP REPOSITORY BACK ENDS   | 5         |
| 1.2. BLOCK STORAGE VOLUME BACKUP METADATA                                | 5         |
| <b>CHAPTER 2. DEPLOYING THE BLOCK STORAGE BACKUP SERVICE</b> .....       | <b>6</b>  |
| 2.1. DEPLOYING YOUR ACTIVE-ACTIVE BLOCK STORAGE BACKUP SERVICE           | 6         |
| 2.2. CHANGING THE DEFAULT BLOCK STORAGE BACKUP SERVICE PARAMETER VALUES  | 7         |
| 2.2.1. Backup repository back-end configuration                          | 7         |
| 2.2.1.1. OpenStack Object Storage service (swift) back end               | 7         |
| 2.2.1.2. NFS back end  | 8         |
| 2.2.1.3. Red Hat Ceph Storage back end                                   | 8         |
| 2.2.1.4. S3 back end   | 8         |
| 2.2.2. Block Storage backup service configuration                        | 9         |
| <b>CHAPTER 3. USING THE BLOCK STORAGE BACKUP SERVICE</b> .....           | <b>10</b> |
| 3.1. CREATING BACKUPS  | 10        |
| 3.1.1. Creating a full volume backup                                     | 10        |
| 3.1.2. Creating a full snapshot backup                                   | 12        |
| 3.1.3. Creating a backup of an in-use volume                             | 13        |
| 3.1.4. Incremental backups   | 15        |
| 3.1.4.1. Creating an incremental backup                                  | 15        |
| 3.1.5. Backup performance considerations                                 | 16        |
| 3.1.6. Backup arguments to authenticate volume owners                    | 17        |
| 3.1.7. Viewing and modifying project backup quotas                       | 17        |
| 3.1.8. Canceling a backup  | 19        |
| 3.2. BACKING UP AND RESTORING ACROSS EDGE SITES                          | 20        |
| 3.3. PROTECTING YOUR BACKUPS   | 21        |
| 3.3.1. Exporting backup metadata   | 21        |
| 3.3.2. Importing backup metadata   | 22        |
| 3.4. RESTORING BACKUPS   | 23        |
| 3.4.1. Restoring a backup to a specific volume                           | 24        |
| 3.4.2. Restoring a backup to a new volume                                | 25        |
| 3.4.3. Canceling restoring a backup                                      | 26        |
| <b>CHAPTER 4. TROUBLESHOOTING THE BLOCK STORAGE BACKUP SERVICE</b> ..... | <b>28</b> |
| 4.1. VERIFYING THE BLOCK STORAGE BACKUP SERVICE DEPLOYMENT               | 28        |
| 4.2. TROUBLESHOOTING BACKUPS   | 28        |
| 4.3. EXAMINING THE BLOCK STORAGE BACKUP SERVICE LOG FILE                 | 29        |
| 4.4. VOLUME BACKUP WORKFLOW  | 29        |
| 4.5. VOLUME RESTORE WORKFLOW   | 31        |



## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

### Providing documentation feedback in Jira

Use the [Create Issue](#) form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.
2. Click the following link to open a the **Create Issue** page: [Create Issue](#)
3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.
4. Click **Create**.



# CHAPTER 1. BLOCK STORAGE BACKUP SERVICE OVERVIEW

The Block Storage service (cinder) of the Red Hat OpenStack Platform (RHOSP) provides an optional backup service that you can deploy on Controller nodes.

You can use the Block Storage backup service to create and restore full or incremental backups of your Block Storage volumes.

A volume backup is a persistent copy of the contents of a Block Storage volume that is saved to a backup repository.

Some features of the Block Storage backup service can impact the performance of the backups. For more information, see [Backup performance considerations](#).

## 1.1. BACKUP REPOSITORY BACK ENDS

By default, your backup repository uses the Red Hat OpenStack Platform Object Storage service (swift) back end and the volume backups are created as object stores. However, you can choose to use Red Hat Ceph Storage, NFS, or S3 as your backup repository back end.

The Block Storage backup service can back up volumes on any back end that the Block Storage service (cinder) supports, regardless of which back end you choose to use for your backup repository.

## 1.2. BLOCK STORAGE VOLUME BACKUP METADATA

When you create a backup of a Block Storage volume, the metadata for this backup is stored in the Block Storage service database. The Block Storage backup service uses this metadata when it restores the volume from the backup.



### IMPORTANT

To ensure that a backup survives a catastrophic loss of the Block Storage service database, you can manually export and store the metadata of this backup. After a catastrophic database loss, you need to create a new Block Storage database and then manually re-import this backup metadata into it. For more information, see [Protecting your backups](#).

## CHAPTER 2. DEPLOYING THE BLOCK STORAGE BACKUP SERVICE

The Block Storage (cinder) backup service is optional. You must include it in your Red Hat OpenStack Platform (RHOSP) overcloud deployment to deploy it on your Controller nodes.

### 2.1. DEPLOYING YOUR ACTIVE-ACTIVE BLOCK STORAGE BACKUP SERVICE

Before Red Hat OpenStack Platform (RHOSP) 17.1, the Block Storage backup service was deployed in active-passive mode and was managed by Pacemaker.

In RHOSP 17.1, the Block Storage backup service is deployed in active-active mode and therefore runs on each Controller node and is not managed by Pacemaker.



#### NOTE

When you upgrade to RHOSP 17.1, the Block Storage backup service remains in active-passive mode.

If you choose to use the Block Storage backup service, then you must include it in your RHOSP 17.1 overcloud deployment.

#### Prerequisites

- An available storage source for your backup repository that uses one of the following back ends: Object Storage (swift), Red Hat Ceph Storage, NFS, or S3.

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. Add this environment file to the stack with your other environment files: **/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup-active-active.yaml**

This file deploys the Block Storage backup service in active-active mode and sets all the heat template parameters of this service to their default settings. The default settings configure your backup repository to use the Object Storage (swift) back end and the **zlib** data compression algorithm.

If the default configuration meets your deployment requirements, then you do not need to do anything further and you can deploy the overcloud.

4. If you need to use another back end for your backup repository or to change other default values:
  - a. Add these parameters and their new values to the **parameter\_defaults** section of either a new or existing environment file. For more information about the parameters that you can change, see [Changing the default Block Storage backup service parameter values](#).

For example, a new environment file `/home/stack/templates/custom_backup_environment_file.yaml` specifies an NFS back end and changes the data compression algorithm to **zstd**:

```
parameter_defaults:
  CinderBackupBackend: nfs
  CinderBackupNfsShare: 192.168.1.1:/var/export/cinder-backup
  CinderBackupCompressionAlgorithm: zstd
```

- b. Add the environment file that contains your specific parameter values to the stack with your other environment files, after the `/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup-active-active.yaml` file and deploy the overcloud. In this example:

```
$ openstack overcloud deploy --templates
-e [your other environment files]
-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup-active-
active.yaml
-e /home/stack/templates/custom_backup_environment_file.yaml
```

#### Verification:

- Ensure that the Block Storage services are running correctly on their hosts and then verify that the Block Storage backup service is deployed successfully. For more information, see [Verifying the Block Storage backup service deployment](#).

## 2.2. CHANGING THE DEFAULT BLOCK STORAGE BACKUP SERVICE PARAMETER VALUES

When you deploy the Block Storage backup service, it implements default settings for its heat template parameters. For more information, see [Deploying your active-active Block Storage backup service](#).

You can provide your deployment specific values for these parameters.

#### Procedure


1. Select and configure the back end for your backup repository. For more information, see [Backup repository back-end configuration](#).
2. Implement the Block Storage backup service configuration supported by your selected back end. For more information, see [Block Storage backup service configuration](#).

### 2.2.1. Backup repository back-end configuration

Select and configure one of the following back ends for your backup repository.

#### 2.2.1.1. OpenStack Object Storage service (swift) back end

| Parameter | Description | Value |
|-----------|-------------|-------|
|-----------|-------------|-------|

|                     |  |   |
|---------------------|--|---|
| CinderBackupBackend | <p>The back end of your backup repository.</p>  <p><b>NOTE</b></p> <p>There are no additional parameters for this default back end.</p> | <p><b>swift</b></p> <p>The default value.</p> |
|---------------------|--|---|

### 2.2.1.2. NFS back end


| Parameter                   | Description  | Value      |
|-----------------------------|--|------------|
| CinderBackupBackend         | The back end of your backup repository.  | <b>nfs</b> |
| CinderBackupNfsShare        | <p>The remote NFS share that you want to mount to store your backups.</p> <p>Ensure that you specify the server name or IP followed by the export.</p> |            |
| CinderBackupNfsMountOptions | Optional: A comma-delimited list of options for mounting your NFS share.   |            |

### 2.2.1.3. Red Hat Ceph Storage back end

| Parameter               | Description  | Value          |
|-------------------------|--|----------------|
| CinderBackupBackend     | The back end of your backup repository.                          | <b>ceph</b>    |
| CinderBackupRbdPoolName | The RBD pool name of your Ceph cluster that stores your backups. | <b>backups</b> |


### 2.2.1.4. S3 back end

| Parameter           | Description                             | Value     |
|---------------------|---|-----------|
| CinderBackupBackend | The back end of your backup repository. | <b>s3</b> |

|                           |   |                      |
|---------------------------|---|----------------------|
| CinderBackupS3Bucket      | The S3 bucket that stores your backups.<br><br> <b>NOTE</b><br><br>Ensure that you create this bucket on the S3 back end and that you have configured the necessary permissions to write to this bucket, before you deploy the Block Storage backup service. | <b>volumebackups</b> |
| CinderBackupS3AccessKey   | The S3 Access key to connect to your S3 bucket.   |                      |
| CinderBackupS3SecretKey   | The S3 Secret key to connect to your S3 bucket.   |                      |
| CinderBackupS3EndpointUrl | The URL of your S3 endpoint.  |                      |

### 2.2.2. Block Storage backup service configuration

You can implement any Block Storage backup service parameter that is supported by your selected back end.

| Parameter                        | Description  | Value   |
|----------------------------------|--|---|
| CinderBackupCompressionAlgorithm | If your back end supports it, you can enable the data compression of your backup repository.<br><br>Data compression requires additional CPU power but uses less network bandwidth and storage space.<br><br> <b>NOTE</b><br><br>You cannot specify the data compression algorithm for the Red Hat Ceph Storage back end driver. This parameter is ignored for this back end. | <b>zlib</b><br><br>Alternatives<br>:<br><br><b>none, bzip2, or zstd</b> |

## CHAPTER 3. USING THE BLOCK STORAGE BACKUP SERVICE

You can use the Block Storage backup service to perform full or incremental backups, to protect your backups, and to restore a backup to a volume.

### 3.1. CREATING BACKUPS

Create a backup of your Block Storage volume to protect your data from being lost if the volume fails. For more information, see [Creating a full volume backup](#). You can also create a backup directly from a snapshot of a volume. For more information, see [Creating a full snapshot backup](#). In addition to the volume data, a backup also stores the volume metadata, such as the name and description.

If you have enabled data compression for your backup repository then your backups will be compressed, which can reduce their performance.

Full backups are easier to manage but they can become resource intensive when the size of the volume increases over time. With incremental backups, you can capture periodic changes to volumes and minimize resource usage. For more information, see [Incremental backups](#).

You can create backups of volumes that you have access to. Project administrators can back up any volume belonging to the project. These backups are hidden from the volume owners unless the administrator provides additional arguments when creating the backup. For more information, see [Backup arguments to authenticate volume owners](#).

Each project (tenant) limits the maximum number of backups and the maximum total size of all backups that can be created for it. As a project administrator, you can view and change these quotas. For more information, see [Viewing and modifying project backup quotas](#).

Normally you can only backup a volume that has an **available** status but, if you need to, you can backup a volume with an **in-use** status. For more information, see [Creating a backup of an in-use volume](#).

When you create a backup of a Block Storage volume, the metadata for this backup is stored in the Block Storage service database, which is used when restoring this volume. To ensure that a backup survives a catastrophic loss of the Block Storage service database, a project administrator can manually export and store the metadata of this backup. For more information, see [Protecting your backups](#).

#### 3.1.1. Creating a full volume backup

You can create one or more full backups of a volume.

##### Prerequisites

- Only volume owners and project administrators can backup volumes.
- Your backup repository must have the necessary space.
- The backup quotas specified for your project have not been exceeded. For more information, see [Viewing and modifying project backup quotas](#).

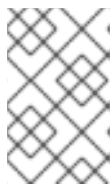
##### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- List the volumes to obtain the ID or name of the volume you want to back up:

```
$ openstack volume list
```



#### NOTE

Usually you can only backup a volume that has an **available** status but, if you need to, you can backup a volume with an **in-use** status. For more information, see [Creating a backup of an in-use volume](#) .

- Back up the volume:



#### NOTE

If you are the project administrator and not the volume owner, then to make this backup accessible by the volume owner, you need to provide additional parameters when creating this backup. For more information, see [Backup arguments to authenticate volume owners](#).

```
$ openstack volume backup create [--name <backup_name>] <volume>
```

- Replace **<volume>** with the ID or name of the volume you want to back up.
- Optional: replace **<backup\_name>** with the name of this backup. This command immediately provides the ID of this backup but the volume is backed up asynchronously, in the background. For example:

```
$ openstack volume backup create --name vol1bu2 vol_1
+-----+-----+-----+-----+
| Field | Value                                |
+-----+-----+-----+-----+
| id    | 83dad43-2aa9-4c0b-bc05-a12203a8f4cb |
| name  | vol1bu2                              |
+-----+-----+-----+-----+
```

### Verification

- List the backups:

```
$ openstack volume backup list
```

The volume backup is created when this backup has an **available** status. For example:

```
+-----+-----+-----+-----+-----+
| ID                | Name   | Description | Status  | Size |
+-----+-----+-----+-----+-----+
| 83dad43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None       | available | 1 |
| b604a932-94a5-468e-bf6b-841ac16f69a8 | None   | None       | available | 1 |
+-----+-----+-----+-----+-----+
```

## Additional resources

- [Troubleshooting backups](#)
- [Creating backups](#)

### 3.1.2. Creating a full snapshot backup

You can create a full backup from a snapshot by using the ID of the volume associated with the snapshot.

The backup is created by directly attaching to the snapshot, which is faster than cloning the snapshot into a volume and then backing up this volume. But this feature can affect the backup performance because of the extra step of creating the volume from a snapshot.

#### Prerequisites:

- Your backup repository must have the necessary space.
- The backup quotas specified for your project have not been exceeded. For more information, see [Viewing and modifying project backup quotas](#).

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. List the snapshots to obtain the name or ID of the snapshot you want to backup:

```
$ openstack volume snapshot list
```

4. List the details of this snapshot to obtain the ID of the volume associated with this snapshot:

```
$ openstack volume snapshot show <snapshot>
```

- Replace **<snapshot>** with the name or ID of the snapshot you want to backup. The value of the **volume\_id** field is the ID of the volume associated with this snapshot. For example:

```
$ openstack volume snapshot show snap_1
+-----+
| Field                | Value                                     |
+-----+-----+
| created_at           | 2023-07-18T08:14:16.000000              |
| description          | None                                     |
| id                   | 2d95b707-6657-49af-ac8f-f9a7417d4650 |
| name                 | snap_1                                  |
| os-extended-snapshot-attributes:progress | 100%                                     |
| os-extended-snapshot-attributes:project_id | c2c1da89ed1648fc8b4f35a045f8d34c |
| properties           |                                           |
| size                 | 1                                        |
+-----+-----+
```



```

| status                | available                |
| updated_at           | 2023-07-18T08:14:17.000000 |
| volume_id            | 6841e3d1-8a1a-4496-bc51-f7c04b787e8f |
+-----+-----+

```

5. Backup the snapshot:

```
$ openstack volume backup create [--name <backup_name>] --snapshot <snapshot>
<volume_id>
```

- Replace **<volume\_id>** with the ID of the volume associated with this snapshot.
- Optional: replace **<backup\_name>** with the name of this backup.  
This command immediately provides the ID of this backup but the snapshot is backed up asynchronously, in the background. For example:

```

$ openstack volume backup create --name snap1bu1 --snapshot snap_1 6841e3d1-
8a1a-4496-bc51-f7c04b787e8f
+-----+-----+
| Field | Value                |
+-----+-----+
| id    | 867e6cfb-9be7-47fa-8a79-221b0e80c757 |
| name  | snap1bu1             |
+-----+-----+

```

## Verification

- List the backups:

```
$ openstack volume backup list
```

The snapshot backup is created when this backup has an **available** status. For example:

```

+-----+-----+-----+-----+-----+
| ID                | Name    | Description | Status  | Size |
+-----+-----+-----+-----+-----+
| 867e6cfb-9be7-47fa-8a79-221b0e80c757 | snap1bu1 | None       | available | 1 |
+-----+-----+-----+-----+-----+

```

### 3.1.3. Creating a backup of an in-use volume

Usually you can only backup a volume that has an **available** status. But you can use the **--force** option when creating a backup, to back up a volume that has an **in-use** status.

When you use the **--force** volume backup option, you create a crash-consistent, but not an application-consistent, backup because the volume is not quiesced before performing the backup. Therefore, the data is intact but the backup does not have an awareness of which applications were running when the backup was performed.

## Prerequisites

- Only volume owners and project administrators can backup volumes.

- Your backup repository must have the necessary space.
- The backup quotas specified for your project have not been exceeded. For more information, see [Viewing and modifying project backup quotas](#).

## Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. List the volumes to obtain the ID or name of the volume you want to back up:

```
$ openstack volume list
```

For example:

```
+-----+-----+-----+-----+-----+
| ID              | Name          | Status | Size | Attached to |
+-----+-----+-----+-----+-----+
| 6841e3d1-8a1a-4496-bc51-f7c04b787e8f | vol_1         | available | 1 |
| 92800cf6-82ae-448a-a2bb-872fa4d98099 | Pansible_vol_2 | in-use   | 1 | Attached to
| inst1 on /dev/vdc |
+-----+-----+-----+-----+-----+
```

4. Force the back up, if the volume that you want to backup has an **in-use** status:

```
$ openstack volume backup create [--name <backup_name>] --force <volume>
```

- Replace **<volume>** with the ID or name of the volume you want to back up.
- Optional: replace **<backup\_name>** with the name of this backup.  
This command immediately provides the ID of this backup but the volume is backed up asynchronously, in the background. For example:

```
$ openstack volume backup create --name panvol2bu1 --force Pansible_vol_2
+-----+-----+-----+
| Field | Value |
+-----+-----+-----+
| id    | 8c72bbf3-eb8e-4459-83e9-c7654ebe6343 |
| name  | panvol2bu1 |
+-----+-----+-----+
```

## Verification

- List the backups:

```
$ openstack volume backup list
```

The volume backup is created when this backup has an **available** status. For example:

```

+-----+-----+-----+-----+
| ID           | Name      | Description | Status  | Size |
+-----+-----+-----+-----+
| 8c72bbf3-eb8e-4459-83e9-c7654ebe6343 | panvol2bu1 | None      | available | 1 |
+-----+-----+-----+-----+

```

### Additional resources

- [Troubleshooting backups](#)
- [Creating backups](#)

## 3.1.4. Incremental backups

If a volume has at least one full backup, you can use the Block Storage backup service to create an incremental backup. For more information, see [Creating an incremental backup](#).

Full backups are easier to manage but they can become resource intensive when the size of the volume increases over time. With incremental backups, you can capture periodic changes to volumes and minimize your resource usage.

An incremental backup only stores the changes made to the volume since the last full or incremental backup.

Incremental backups increase the administrative overhead required for managing your backups. For instance, you cannot delete a full backup if it already has one or more incremental backups, you can only delete the latest incremental backup.

Incremental backups have a lower performance than full backups: When you create an incremental backup, all of the data in the volume must first be read and compared with the data in both the full backup and each subsequent incremental backup.

### 3.1.4.1. Creating an incremental backup

You can create an incremental backup to only store the changes made to the volume since the last full or incremental backup.

#### Prerequisites:

- At least one full backup of the volume. For more information, see [Creating a full volume backup](#).
- Only volume owners and project administrators can backup volumes.
- Your backup repository must have the necessary space.
- The backup quotas specified for your project have not been exceeded. For more information, see [Viewing and modifying project backup quotas](#).

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- 
- 3. List the volumes to obtain the ID or name of the volume you want to back up:

```
$ openstack volume list
```

- 4. Back up the volume and use the **--incremental** option:

```
$ openstack volume backup create --incremental [--name <backup_name>] <volume>
```

- Replace **<volume>** with the ID or name of the volume you want to back up.
- Optional: replace **<backup\_name>** with the name of this backup. This command immediately provides the ID of this backup but the volume is backed up asynchronously, in the background. For example:

```
$ openstack volume backup create --name vol3incbu1 --incremental vol_3
+-----+-----+-----+-----+
| Field | Value                |
+-----+-----+-----+-----+
| id    | f1681313-b5ed-4520-9b63-5b533f7cdc11 |
| name  | vol3incbu1           |
+-----+-----+-----+-----+
```

## Verification

- List the backups:

```
$ openstack volume backup list
```

The volume backup is created when this backup has an **available** status. For example:

```
+-----+-----+-----+-----+-----+
| ID                | Name   | Description | Status  | Size |
+-----+-----+-----+-----+-----+
| f1681313-b5ed-4520-9b63-5b533f7cdc11 | vol3incbu1 | None       | available | 1 |
| f0e9ba67-67e1-4c2c-96ce-221df75bf2c2 | vol3bu1   | None       | available | 1 |
+-----+-----+-----+-----+-----+
```

## Additional resources

- [Troubleshooting backups](#)

### 3.1.5. Backup performance considerations

Some features of the Block Storage backup service like incremental backups and data compression can reduce the performance of the backups.

By only capturing the periodic changes to volumes, incremental backups can minimize your resource usage. For more information, see [Incremental backups](#). But incremental backups have a lower performance than full backups: When you create an incremental backup, all of the data in the volume must first be read and compared with the data in both the full backup and each subsequent incremental backup.

You can also create a backup from a snapshot by directly attaching to it, which is faster than cloning the snapshot into a volume. For more information, see [Creating a full snapshot backup](#). But this feature can affect the backup performance because of the extra step of creating the volume from a snapshot.

Enabling data compression of your backup repository requires additional CPU power but uses less network bandwidth and storage space overall. You can configure the Block Storage backup service to enable or disable data compression of your backup repository. For more information, see [Block Storage backup service configuration](#).



#### NOTE

You cannot use data compression with the Red Hat Ceph Storage back end.

### 3.1.6. Backup arguments to authenticate volume owners

Project administrators can back up any volume belonging to the project, but these backups are hidden from the volume owners.

To ensure that the volume owner can also access the volume backup, the project administrator must provide the following additional arguments to authenticate as the volume owner when backing up the volume:

```
$ openstack --os-auth-url <keystoneurl> --os-project-name <projectname> --os-username
<username> --os-password <password> volume backup create [--name <backup_name>] <volume>
```

- Replace **<keystoneurl>** with the URL endpoint of the Identity service, which is typically <http://IP:5000/v3>, where **IP** is the IP address of the Identity service host.
- Replace **<projectname>** with the name of the project (tenant) of the owner of the volume.
- Replace **<username>** and **<password>** with the username and password credentials of the user that is the owner of the volume within this project.



#### NOTE

**[--name <backup\_name>] <volume>** are the typical arguments when creating a volume backup.

### 3.1.7. Viewing and modifying project backup quotas

A project administrator can change or view the maximum number of backups and the maximum total size of all backups, in gigabytes, that can be created for a specific project (tenant) and see the usage of these backup quotas for this project.

#### Prerequisites

- You must be a project administrator to view or modify the backup quotas of a project.

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- List the projects to obtain the ID or name of the required project:

```
$ openstack project list
```

- View the backup quotas for a specific project:

```
$ openstack quota show <project>
```

- Replace **<project>** with the ID or name of the required project.

The value of the **backup-gigabytes** field in the table is the maximum total size of all backups that can be created in this project. The value of the **backups** field in the table is the maximum number of backups that can be created in this project. For example:

+

```
$ openstack quota show c2c1da89ed1648fc8b4f35a045f8d34c
```

```
+-----+-----+-----+-----+
| Field          | Value |
|-----+-----+-----+-----+
| backup-gigabytes | 1000 |
|-----+-----+-----+-----+
| backups         | 10    |
|-----+-----+-----+-----+
```

- Modify the maximum total size of all backups created for a project:

```
$ openstack quota set --backup-gigabytes <maxgb> <project>
```

- Replace **<maxgb>** with the maximum total size, in gigabytes, of the backups that can be created for this project.

- Modify the maximum number of backups that can be created for a project:

```
$ openstack quota set --backups <maxnum> <project>
```

- Replace **<maxnum>** with the maximum number of backups that can be created for this project.

- View the usage of these backup quotas for a specific project:

```
$ cinder quota-usage <project_id>
```

- Replace **<project\_id>** with the ID of the project.  
For example:

```
$ cinder quota-usage c2c1da89ed1648fc8b4f35a045f8d34c
+-----+-----+-----+-----+-----+
| Type          | In_use | Reserved | Limit | Allocated |
|-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+
| backup_gigabytes | 7 | 0 | 1000 | |
| backups          | 7 | 0 | 10 | |
| gigabytes        | 6 | 0 | 1000 | |
| gigabytes_multiattach | 0 | 0 | -1 | |
| gigabytes_tripleo | 6 | 0 | -1 | |
| groups           | 0 | 0 | 10 | |
| per_volume_gigabytes | 0 | 0 | -1 | |
| snapshots        | 1 | 0 | 10 | |
| snapshots_multiattach | 0 | 0 | -1 | |
| snapshots_tripleo | 1 | 0 | -1 | |
| volumes          | 5 | 0 | 10 | |
| volumes_multiattach | 0 | 0 | -1 | |
| volumes_tripleo  | 5 | 0 | -1 | |
+-----+-----+-----+-----+

```

## Verification

- If you have changed either of these quotas then review these changes:

```
$ openstack quota show <project>
```

Ensure that the modified values are specified by the **backup-gigabytes** and **backups** fields in the table. For example:

```

+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Field          | Value |
+-----+-----+-----+-----+
| backup-gigabytes | 500 |
| backups         | 12 |
+-----+-----+-----+-----+

```

### 3.1.8. Canceling a backup

You must request a force delete on the backup to cancel it.



#### IMPORTANT

You cannot cancel backups if you use the Red Hat Ceph Storage back end for your backup repository.

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. List the backups to obtain the ID or name of the backup you want to cancel:

-

```
$ openstack volume backup list
```

- Cancel the backup:

```
# openstack volume backup delete --force <backup>
```

- Replace **<backup>** with the ID or name of the volume backup that you want to cancel. There can be a slight delay for the backup to be successfully canceled.

### Verification

- The backup is canceled when the backup record is not listed by this command:

```
$ openstack volume backup show <backup>
```

## 3.2. BACKING UP AND RESTORING ACROSS EDGE SITES

You can back up and restore Block Storage service (cinder) volumes across distributed compute node (DCN) architectures in edge site and availability zones. The **cinder-backup** service runs in the central availability zone (AZ), and backups are stored in the central AZ. The Block Storage service does not store backups at DCN sites.

### Prerequisites

- Deploy the optional Block Storage backup service. For more information, see [Block Storage backup service deployment](#) in *Backing up Block Storage volumes*.
- Block Storage (cinder) REST API microversion 3.51 or later.
- All sites must use a common **openstack** cephx client name. For more information, see [Creating a Ceph key for external access](#) in *Deploying a Distributed Compute Node (DCN) architecture*.

### Procedure

- Create a backup of a volume in the first DCN site:

```
$ cinder --os-volume-api-version 3.51 backup-create --name <volume_backup> --availability-zone <az_central> <edge_volume>
```

- Replace **<volume\_backup>** with a name for the volume backup.
- Replace **<az\_central>** with the name of the central availability zone that hosts the **cinder-backup** service.
- Replace **<edge\_volume>** with the name of the volume that you want to back up.



#### NOTE

If you experience issues with Ceph keyrings, you might need to restart the **cinder-backup** container so that the keyrings copy from the host to the container successfully.

- Restore the backup to a new volume in the second DCN site:



```
$ cinder --os-volume-api-version 3.51 create --availability-zone <az_2> --name
<new_volume> --backup-id <volume_backup> <volume_size>
```

- Replace **<az\_2>** with the name of the availability zone where you want to restore the backup.
- Replace **<new\_volume>** with a name for the new volume.
- Replace **<volume\_backup>** with the name of the volume backup that you created in the previous step.
- Replace **<volume\_size>** with a value in GB equal to or greater than the size of the original volume.

### 3.3. PROTECTING YOUR BACKUPS

When you create a backup of a Block Storage volume, the metadata for this backup is stored in the Block Storage service database, which is used to restore this volume. To ensure that a backup survives a catastrophic loss of the Block Storage service database, a project administrator can manually export and store the metadata of this backup in a safe location, such as an offsite backup. For more information, see [Exporting backup metadata](#).

When the Block Storage service database experiences a catastrophic loss you cannot restore any of your backups because this database contains the backup metadata used when restoring backups. But if a project administrator manually exported and saved the metadata of a backup, then the project administrator can import this metadata into the new Block Storage database, so that you can use this backup to restore the volume. For more information see [Importing backup metadata](#).



#### NOTE

For incremental backups, you must import the metadata of all the preceding backups before you can use it to restore the volume.

#### 3.3.1. Exporting backup metadata

A project administrator can export the metadata of a backup and store it in a file so that you can restore the volume backup even if the Block Storage database suffers a catastrophic loss. For more information, see [Protecting your backups](#).



#### NOTE

For an incremental backup, you must export the metadata of all the preceding backups.

#### Prerequisites

- You must be a project administrator to export the backup metadata.

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- List the backups to obtain the ID or name of the backup:

```
$ openstack volume backup list
```

- Export the metadata of the backup and store it in an appropriately named YAML file:

```
$ openstack volume backup record export -f yaml <backup> > <filename>.yaml
```

- Replace **<backup>** with the ID or name of the volume backup.
- Replace **<filename>** with the name of the YAML file to save the exported **backup\_service** and **backup\_url** values for this backup.  
For example:

```
$ openstack volume backup record export -f yaml vol1bu2 > vol1bu2.yaml
```

- Copy the file to a safe location, such as an offsite backup.

### Verification

- Edit the file to see that the values of the **backup\_service** and **backup\_url**, match the values provided by this command:

```
$ openstack volume backup record export -f yaml <backup>
```

For example:

```
$ openstack volume backup record export -f yaml vol1bu2
backup_service: cinder.backup.drivers.ceph.CephBackupDriver
backup_url: eyJkcml2 ... YWxzZX0=
```

### 3.3.2. Importing backup metadata

If a project administrator has exported and saved the metadata of a volume backup, then after a catastrophic loss of the Block Storage service database, the project administrator can import this metadata, so that you can use this backup.

You can also use this procedure to recreate a backup that was deleted.



#### NOTE

For incremental backups, you must also import the metadata of all the preceding backups.

### Prerequisites

- You must be a project administrator to import the volume backup metadata to a Block Storage database.
- You must provide the **backup\_service** and **backup\_url** metadata values of this backup. For more information see [Exporting backup metadata](#).
- A Block Storage database that does not already contain this backup.

## Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. Locate the file in which you stored the exported **backup\_service** and **backup\_url** metadata values of this backup.
4. Import the metadata values of this volume backup to the Block Storage database:

```
$ openstack volume backup record import <backup_service> <backup_url>
```

- Replace **<backup\_service>** with the **backup\_service** metadata value of this volume backup.
- Replace **<backup\_url>** with the **backup\_url** metadata value of this volume backup. This command provides the name and the ID of this backup. For example:

```
$ openstack volume backup record import cinder.backup.drivers.ceph.CephBackupDriver
eyJkcml2 ... YWxzZX0=
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 83dad43-2aa9-4c0b-bc05-a12203a8f4cb |
| name  | vol1bu2 |
+-----+-----+
```

## Next steps

- [Restoring backups](#)

## 3.4. RESTORING BACKUPS

After you create a Block Storage volume backup, you can restore this backed up data if needed.

You can use one of the following methods to restore your backups:

- Restore the backup to a volume that you specify. For more information, see [Restoring a backup to a specific volume](#).
- Restore the backup to a new volume. For more information, see [Restoring a backup to a new volume](#).



### IMPORTANT

When the Block Storage service database experiences a catastrophic loss you cannot restore any of your backups, unless you have exported and saved their metadata. For more information see [Protecting your backups](#).

Only project administrators can cancel restoring a volume backup. For more information, see [Canceling restoring a backup](#).

### 3.4.1. Restoring a backup to a specific volume

You can restore a volume backup to an **available** volume that you have already created.

If you restore a volume from an encrypted backup, then the destination volume type must also be encrypted.

#### Procedure

1. Log in to the undercloud host as the **stack** user.
2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. List the backups to obtain the name or ID of the backup you want to restore:

```
$ openstack volume backup list
```

For example:

```
+-----+-----+-----+-----+-----+
| ID                | Name  | Description | Status | Size |
+-----+-----+-----+-----+-----+
| 83dad43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None      | available | 1 |
```

4. List the volumes:

```
$ openstack volume list
```

Ensure that the status of the required volume is **available** and then obtain the name or ID of this volume. For example:

```
+-----+-----+-----+-----+-----+
| ID                | Name  | Status | Size | Attached to |
+-----+-----+-----+-----+-----+
| 654e2be8-bc79-4528-96a7-5f773d31c201 | vol_3 | available | 1 |
```

5. Restore the backup to the volume:

```
$ openstack volume backup restore <backup> <volume>
```

- Replace **<backup>** with the name or ID of the Block Storage volume backup.
  - Replace **<volume>** with the name or ID of the **available** Block Storage volume.
- For example:

```
$ openstack volume backup restore vol1bu2 vol_3
+-----+-----+-----+
| Field  | Value |
+-----+-----+
| backup_id | 83dad43-2aa9-4c0b-bc05-a12203a8f4cb |
```

```
| volume_id | 654e2be8-bc79-4528-96a7-5f773d31c201 |
| volume_name | vol_3 |
+-----+-----+-----+-----+
```

- Verify that the **backup\_id** provided by this command corresponds to the ID of the backup that was restored and that the **volume\_name** and **volume\_id** values correspond to the name and ID of the specified volume.
- Delete the backup if you no longer need it:

```
$ openstack volume backup delete <backup>
```

### 3.4.2. Restoring a backup to a new volume

You can create a new volume when you restore a backup of a Block Storage volume.

#### Procedure

- Log in to the undercloud host as the **stack** user.
- Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- List the backups to obtain the name or ID of the backup you want to restore:

```
$ openstack volume backup list
```

For example:

```
+-----+-----+-----+-----+-----+
| ID              | Name  | Description | Status | Size |
+-----+-----+-----+-----+-----+
| 83dad43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None      | available | 1 |
```

- Restore the backup to a new volume:

```
$ cinder backup-restore <backup>
```

- Replace **<backup>** with the name or ID of the Block Storage volume backup.  
For example:

```
$ cinder backup-restore vol1bu2
+-----+-----+-----+-----+
| Property | Value |
+-----+-----+-----+-----+
| backup_id | 83dad43-2aa9-4c0b-bc05-a12203a8f4cb |
| volume_id | 296c853c-c749-4eb6-857a-57ec182232a6 |
| volume_name | restore_backup_83dad43-2aa9-4c0b-bc05-a12203a8f4cb |
+-----+-----+-----+-----+
```

- Verify that the **backup\_id** provided by this command corresponds to the ID of the backup that was restored.

The **volume\_id** value is the ID of the created volume. But the **volume\_name** can be a temporary name that is replaced with the name of the backed up volume.

- List the volumes to verify that the volume with an ID of **volume\_id** has been created and to obtain this volume name:

```
$ openstack volume list
```

For example:

```
+-----+-----+-----+-----+-----+
| ID              | Name      | Status | Size | Attached to |
+-----+-----+-----+-----+-----+
| 296c853c-c749-4eb6-857a-57ec182232a6 | vol_1     | available | 1 |
```

- Delete the backup if you no longer need it:

```
$ openstack volume backup delete <backup>
```

### 3.4.3. Canceling restoring a backup

A project administrator can cancel restoring a volume backup by changing the status of the backup to **error**. But you cannot cancel restoring a backup when Red Hat Ceph Storage is the back end of the backup repository.



#### WARNING

If you cancel restoring a backup after it starts, the destination volume is useless, because there is no way of knowing how much data, if any, was actually restored.

#### Prerequisites

- You must be a project administrator to cancel restoring a volume backup.
- Ensure that the back end of your backup repository is not Red Hat Ceph Storage.

#### Procedure

- Log in to the undercloud host as the **stack** user.
- Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

- List the backups to obtain the name or ID of the backup that you want to stop restoring:

```
$ openstack volume backup list
```

4. Change the status of this backup to **error** to cancel its restore operation:

```
┆ $ openstack volume backup set --state error <backup>
```

- Replace **<backup>** with the name or ID of the volume backup that you do not want to restore.

Canceling a restore is an asynchronous action, because the back end of the backup repository must detect the change in the backup status before it cancels the restore.

### Verification

- List the volume backups to verify that the restore is canceled:

```
┆ $ openstack volume backup list
```

When the status of the backup changes to **available**, then the restore is canceled.

## CHAPTER 4. TROUBLESHOOTING THE BLOCK STORAGE BACKUP SERVICE

You can diagnose many issues by verifying that the Block Storage services are running correctly and then by examining the log files for error messages.

### 4.1. VERIFYING THE BLOCK STORAGE BACKUP SERVICE DEPLOYMENT

After a deployment or when troubleshooting issues, it is important to verify that the necessary Block Storage services are running correctly on their hosts. Ensure that the Block Storage backup service is running on every Controller node, like the Block Storage scheduler service.

After you verify that the necessary Block Storage services are running correctly, then you must verify that the Block Storage backup service is deployed successfully.

#### Procedure

1. Run the **openstack volume service list** command:

```
# openstack volume service list
+-----+-----+-----+-----+-----+-----+
| Binary      | Host           | Zone | Status | State | Updated At           |
+-----+-----+-----+-----+-----+-----+
| cinder-scheduler | controller-0   | nova | enabled | up   | 2023-06-21T13:07:42.000000 |
| cinder-scheduler | controller-1   | nova | enabled | up   | 2023-06-21T13:07:42.000000 |
| cinder-scheduler | controller-2   | nova | enabled | up   | 2023-06-21T13:07:42.000000 |
| cinder-backup    | controller-0   | nova | enabled | up   | 2023-06-21T13:07:46.000000 |
| cinder-backup    | controller-1   | nova | enabled | up   | 2023-06-21T13:07:46.000000 |
| cinder-backup    | controller-2   | nova | enabled | up   | 2023-06-21T13:07:46.000000 |
| cinder-volume    | hostgroup@tripleo_iscsi | nova | enabled | up   | 2023-06-21T13:07:47.000000 |
+-----+-----+-----+-----+-----+-----+
```

2. Verify that the **State** entry of every service is **up**. If not, examine the relevant log files. For more information about the location of these log files, see [Block Storage \(cinder\) Log Files](#) in *Managing overcloud observability*.
3. Verify that the Block Storage backup service is deployed successfully, by backing up any Block Storage volume and ensuring that the backup succeeds. For more information, see [Troubleshooting backups](#).

#### Additional resources

- [Examining the Block Storage backup service log file](#)

### 4.2. TROUBLESHOOTING BACKUPS

The Block Storage backup service performs static checks when receiving a request to back up a Block Storage (cinder) volume. If these checks fail then you will immediately be notified:



- Check for an invalid volume reference (**missing**).
- Check if the volume is **in-use** or attached to an instance. The **in-use** case requires you to use the **--force** option to perform a backup. For more information, see [Creating a backup of an in-use volume](#).

When you use the **--force** volume backup option, you create a crash-consistent, but not an application-consistent, backup because the volume is not quiesced before performing the backup. Therefore, the data is intact but the backup does not have an awareness of which applications were running when the backup was performed.

When these checks succeed: the Block Storage backup service accepts the request to backup this volume, the CLI backup command returns immediately, and the volume is backed up in the background.

Therefore the CLI backup command returns even if the backup fails. You can use the **openstack volume backup list** command to verify that the volume backup is successful, when the **Status** of the backup entry is **available**.

If a backup fails, examine the Block Storage backup service log file for error messages to discover the cause. For more information, see [Examining the Block Storage backup service log file](#) .

### 4.3. EXAMINING THE BLOCK STORAGE BACKUP SERVICE LOG FILE

When a backup or restore does not succeed, you can examine the Block Storage backup service log file for error messages that can help you to determine the reason.

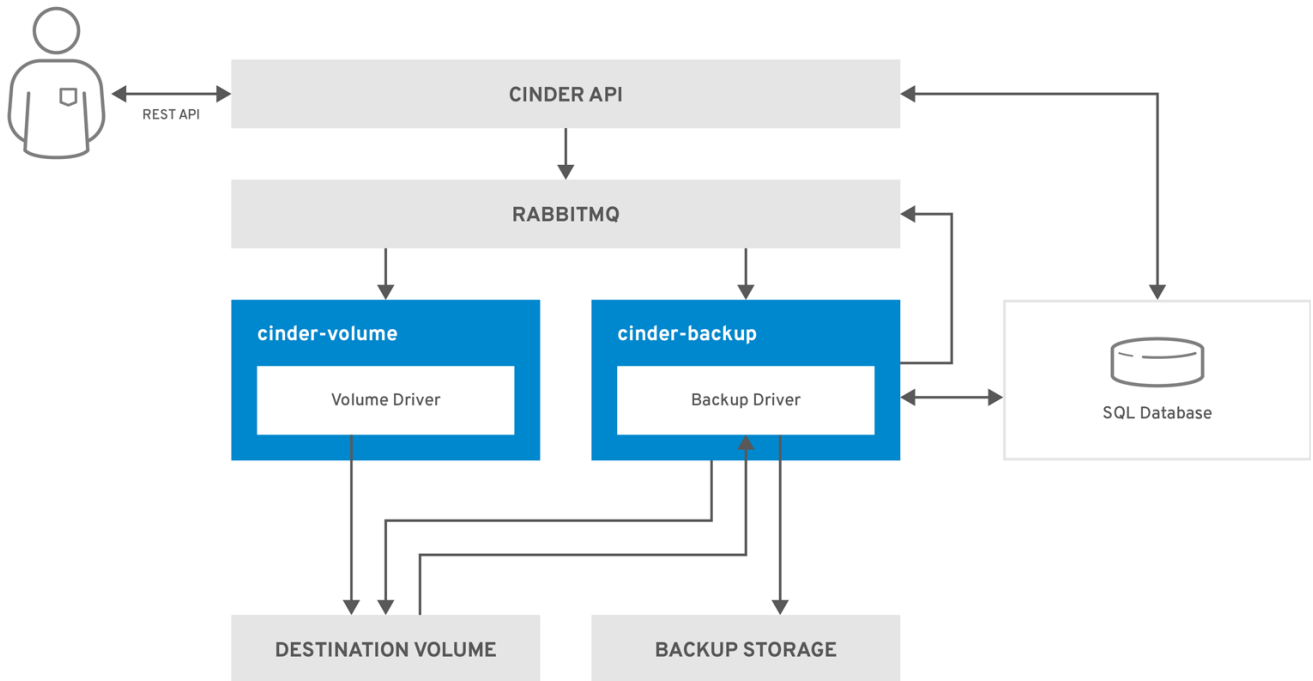
#### Procedure

- Find the Block Storage backup service log file on the Controller node where the backup service is running.  
This log file is located in the following path: **/var/log/containers/cinder/cinder-backup.log**.

### 4.4. VOLUME BACKUP WORKFLOW

The following diagram and explanation describe the steps that occur when the user requests the cinder API to backup a Block Storage (cinder) volume.

Figure 4.1. Creating a backup of a Block Storage volume



OPENSTACK\_483337\_1218

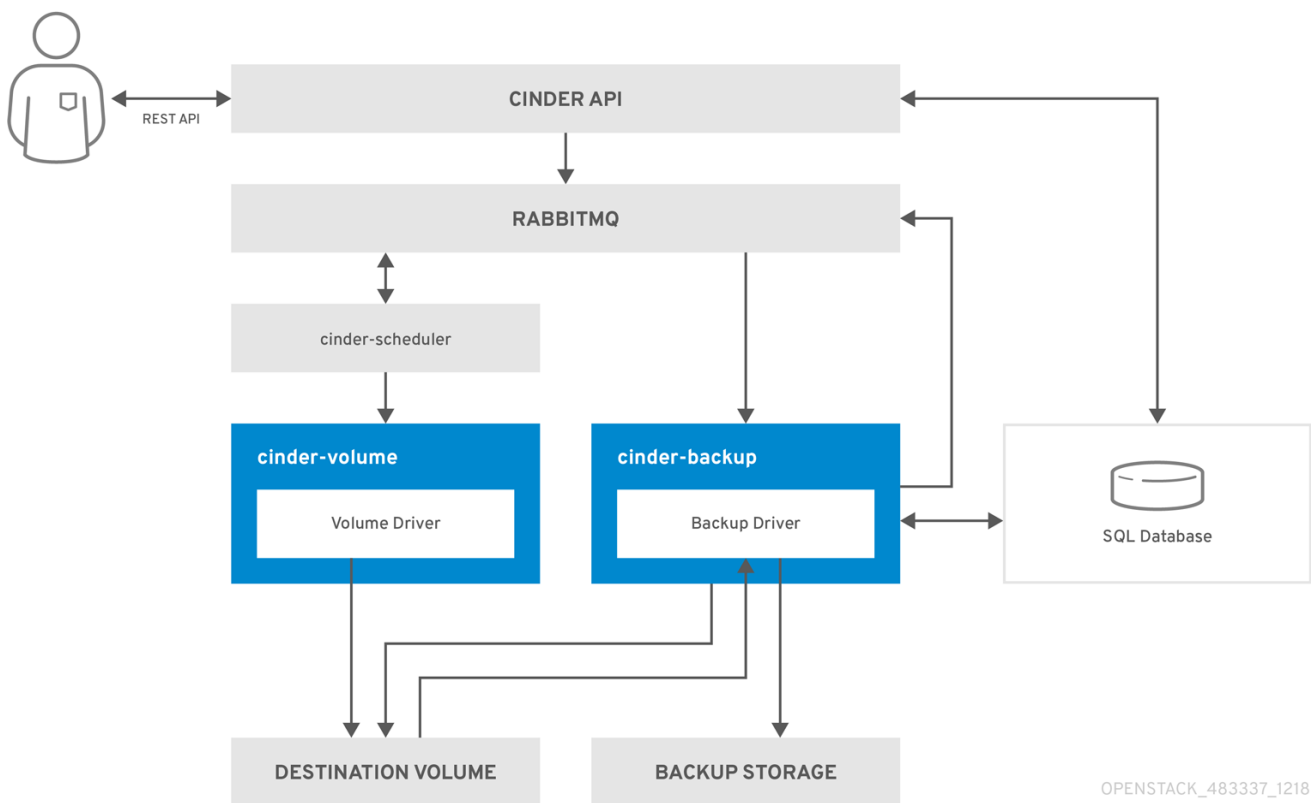
1. The user issues a request to the cinder API, which is a REST API, to back up a Block Storage volume.
2. The cinder API receives the request from HAProxy and validates the request, the user credentials, and other information.
3. The cinder API creates the backup record in the SQL database.
4. The cinder API makes an asynchronous RPC call to the **cinder-backup** service via AMQP to back up the volume.
5. The cinder API returns the current backup record, with an ID, to the API caller.
6. An RPC create message arrives on one of the backup services.
7. The **cinder-backup** service performs a synchronous RPC call to **get\_backup\_device**.
8. The **cinder-volume** service ensures that the correct device is returned to the caller. Normally, it is the same volume, but if the volume is in use, the service returns a temporary cloned volume or a temporary snapshot, depending on the configuration.
9. The **cinder-backup** service issues another synchronous RPC to **cinder-volume** to expose the source device.
10. The **cinder-volume** service exports and maps the source device (volume or snapshot) and returns the appropriate connection information.
11. The **cinder-backup** service attaches the source device by using the connection information.
12. The **cinder-backup** service calls the backup back end driver, with the device already attached, which begins the data transfer to the backup repository.
13. The source device is detached from the Backup host.

14. The **cinder-backup** service issues a synchronous RPC to **cinder-volume** to disconnect the source device.
15. The **cinder-volume** service unmaps and removes the export for the device.
16. If a temporary volume or temporary snapshot was created, **cinder-backup** calls **cinder-volume** to remove it.
17. The **cinder-volume** service removes the temporary volume.
18. When the backup is completed, the backup record is updated in the database.

## 4.5. VOLUME RESTORE WORKFLOW

The following diagram and explanation describe the steps that occur when the user requests the cinder API to restore a Block Storage service (cinder) backup.

Figure 4.2. Restoring a Block Storage backup



1. The user issues a request to the cinder API, which is a REST API, to restore a Block Storage backup.
2. The cinder API receives the request from HAProxy and validates the request, the user credentials, and other information.
3. If the request does not contain an existing volume as the destination, the cinder API makes an asynchronous RPC call to create a new volume and polls the status of the volume until it becomes available.
4. The **cinder-scheduler** selects a volume service and makes the RPC call to create the volume.
5. The selected **cinder-volume** service creates the volume.

6. When the cinder API detects that the volume is available, the backup record is created in the database.
7. The cinder API makes an asynchronous RPC call to the backup service via AMQP to restore the backup.
8. The cinder API returns the current volume ID, backup ID, and volume name to the API caller.
9. An RPC create message arrives on one of the backup services.
10. The **cinder-backup** service performs a synchronous RPC call to **cinder-volume** to expose the volume.
11. The **cinder-volume** service exports and maps the volume returning the appropriate connection information.
12. The **cinder-backup** service attaches the volume by using the connection information.
13. The **cinder-backup** service calls the back end driver with the volume already attached, which begins the data restoration to the volume.
14. The volume is detached from the backup host.
15. The **cinder-backup** service issues a synchronous RPC to **cinder-volume** to disconnect the volume.
16. The **cinder-volume** service unmaps and removes the export for the volume.
17. When the volume is restored, the backup record is updated in the database.