



Red Hat OpenStack Platform 17.1

블록 스토리지 볼륨 백업

Red Hat OpenStack Platform Block Storage(cinder) 백업 서비스 배포 및 사용

Red Hat OpenStack Platform 17.1 블록 스토리지 볼륨 백업

Red Hat OpenStack Platform Block Storage(cinder) 백업 서비스 배포 및 사용

OpenStack Team
rhos-docs@redhat.com

법적 공지

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

초록

Red Hat OpenStack Platform 블록 스토리지 백업 서비스를 배포하고 사용하여 블록 스토리지 볼륨을 백업 및 복원합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
RED HAT 문서에 관한 피드백 제공	4
1장. 블록 스토리지 백업 서비스 개요	5
1.1. 백업 리포지토리 백엔드	5
1.2. 블록 스토리지 볼륨 백업 메타데이터	5
2장. 블록 스토리지 백업 서비스 배포	6
2.1. ACTIVE-ACTIVE BLOCK STORAGE 백업 서비스 배포	6
2.2. 기본 블록 스토리지 백업 서비스 매개변수 값 변경	7
3장. 블록 스토리지 백업 서비스 사용	10
3.1. 백업 생성	10
3.2. 에지 사이트 간 백업 및 복원	20
3.3. 백업 보호	22
3.4. 백업 복원	25
4장. 블록 스토리지 백업 서비스 문제 해결	32
4.1. 블록 스토리지 백업 서비스 배포 확인	32
4.2. 백업 문제 해결	33
4.3. 블록 스토리지 백업 서비스 로그 파일 검사	33
4.4. 볼륨 백업 워크플로	34
4.5. 볼륨 복원 워크플로	36

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

Jira에서 문서 피드백 제공

문제 생성 양식을 사용하여 문서에 대한 피드백을 제공합니다. Jira 문제는 Red Hat OpenStack Platform Jira 프로젝트에서 생성되어 피드백의 진행 상황을 추적할 수 있습니다.

1. Jira에 로그인했는지 확인합니다. Jira 계정이 없는 경우 피드백을 제출할 계정을 생성합니다.
2. 다음 링크를 클릭하여 **문제 생성** 페이지를 엽니다.
<https://issues.redhat.com/secure/CreateInfoDetails!init.jspx?pid=12336920&summary=Documentation%20feedback:%20%3CAdd%20summary%20here%3E&i<Include+the+documentation+URL,+the%20chapter+or+section+number,+and+a+detailed+descrip>
3. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.
4. **생성**을 클릭합니다.

1장. 블록 스토리지 백업 서비스 개요

RHOSP(Red Hat OpenStack Platform)의 Block Storage 서비스(cinder)는 컨트롤러 노드에 배포할 수 있는 선택적 백업 서비스를 제공합니다.

Block Storage 백업 서비스를 사용하여 블록 스토리지 볼륨의 전체 또는 증분 백업을 생성하고 복원할 수 있습니다.

볼륨 백업은 백업 리포지토리에 저장된 Block Storage 볼륨의 영구적인 사본입니다.

블록 스토리지 백업 서비스의 일부 기능은 백업 성능에 영향을 미칠 수 있습니다. 자세한 내용은 [백업 성능 고려 사항](#)을 참조하십시오.

1.1. 백업 리포지토리 백엔드

기본적으로 백업 리포지토리에서 Red Hat OpenStack Platform Object Storage 서비스(swift) 백엔드를 사용하고 볼륨 백업은 오브젝트 저장소로 생성됩니다. 그러나 Red Hat Ceph Storage, NFS 또는 S3을 백업 리포지토리 백엔드로 사용하도록 선택할 수 있습니다.

블록 스토리지 백업 서비스는 백업 리포지토리에 사용하도록 선택한 백엔드에 관계없이 Block Storage 서비스(cinder)에서 지원하는 백엔드의 볼륨을 백업할 수 있습니다.

1.2. 블록 스토리지 볼륨 백업 메타데이터

블록 스토리지 볼륨의 백업을 생성할 때 이 백업의 메타데이터는 블록 스토리지 서비스 데이터베이스에 저장됩니다. 블록 스토리지 백업 서비스는 백업에서 볼륨을 복원할 때 이 메타데이터를 사용합니다.



중요

백업이 블록 스토리지 서비스 데이터베이스의 치명적인 손실 유지되도록 하려면 이 백업의 메타데이터를 수동으로 내보내고 저장할 수 있습니다. 치명적인 데이터베이스 손실 후 새 블록 스토리지 데이터베이스를 만든 다음 수동으로 이 백업 메타데이터를 다시 가져와야 합니다. 자세한 내용은 [백업](#)을 참조하십시오.

2장. 블록 스토리지 백업 서비스 배포

Block Storage(cinder) 백업 서비스는 선택 사항입니다. 이를 컨트롤러 노드에 배포하려면 RHOSP(Red Hat OpenStack Platform) 오버클라우드 배포에 포함해야 합니다.

2.1. ACTIVE-ACTIVE BLOCK STORAGE 백업 서비스 배포

RHOSP(Red Hat OpenStack Platform) 17.1 이전에는 Block Storage 백업 서비스가 활성-패시브 모드로 배포되었으며 Pacemaker에서 관리했습니다.

RHOSP 17.1에서 Block Storage 백업 서비스는 active-active 모드로 배포되므로 각 컨트롤러 노드에서 실행되며 Pacemaker에서 관리하지 않습니다.



참고

RHOSP 17.1로 업그레이드할 때 블록 스토리지 백업 서비스는 active-passive 모드로 유지됩니다.

Block Storage 백업 서비스를 사용하도록 선택하는 경우 RHOSP 17.1 오버클라우드 배포에 포함해야 합니다.

사전 요구 사항

- 다음 백엔드 중 하나를 사용하는 백업 리포지토리에 사용 가능한 스토리지 소스: Object Storage(swift), Red Hat Ceph Storage, NFS 또는 S3.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 다른 환경 파일을 사용하여 스택에 이 환경 파일을 추가합니다. **/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup-active-active.yaml**

이 파일은 Block Storage 백업 서비스를 active-active 모드로 배포하고 이 서비스의 모든 heat 템플릿 매개변수를 기본 설정으로 설정합니다. 기본 설정은 Object Storage(swift) 백엔드 및 **zlib** 데이터 압축 알고리즘을 사용하도록 백업 리포지토리를 구성합니다.

기본 구성이 배포 요구 사항을 충족하는 경우 더 이상 수행할 필요가 없으며 오버클라우드를 배포할 수 있습니다.

4. 백업 리포지토리에 다른 백엔드를 사용하거나 다른 기본값을 변경해야 하는 경우:
 - a. 이러한 매개변수와 해당 새 값을 새 환경 파일의 **parameter_defaults** 섹션에 추가합니다. 변경할 수 있는 매개변수에 대한 자세한 내용은 [기본 블록 스토리지 백업 서비스 매개변수 값](#) 변경을 참조하십시오.
예를 들어 새 환경 파일 **/home/stack/templates/custom_backup_environment_file.yaml**은 NFS 백엔드를 지정하고 데이터 압축 알고리즘을 **zstd**로 변경합니다.

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

CinderBackupCompressionAlgorithm: zstd

- b. `/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup-active-active.yaml` 파일과 함께 특정 매개변수 값이 포함된 환경 파일을 스택에 추가하고 오버클라우드를 배포합니다. 이 예제에서는 다음을 수행합니다.

```
$ 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
```

확인:

- 블록 스토리지 서비스가 호스트에서 올바르게 실행 중인지 확인한 다음 블록 스토리지 백업 서비스가 성공적으로 배포되었는지 확인합니다. 자세한 내용은 [블록 스토리지 백업 서비스 배포 확인](#)을 참조하십시오.

2.2. 기본 블록 스토리지 백업 서비스 매개변수 값 변경

Block Storage 백업 서비스를 배포할 때 해당 heat 템플릿 매개변수에 대한 기본 설정을 구현합니다. 자세한 내용은 [active-active Block Storage 백업 서비스 배포](#)를 참조하십시오.

이러한 매개변수에 대한 배포별 값을 제공할 수 있습니다.

프로세스

- 백업 리포지토리의 백엔드를 선택하고 구성합니다. 자세한 내용은 [Backup repository back-end 구성](#)을 참조하십시오.
- 선택한 백엔드에서 지원하는 블록 스토리지 백업 서비스 구성을 구현합니다. 자세한 내용은 [Block Storage 백업 서비스 구성](#)을 참조하십시오.

2.2.1. 리포지토리 백엔드 구성 백업

백업 리포지토리에 대해 다음 백엔드 중 하나를 선택하고 구성합니다.

2.2.1.1. OpenStack Object Storage 서비스(swift) 백엔드

매개변수	설명	현재의
CinderBackupBackend	백업 리포지토리의 백엔드입니다.  참고 이 기본 백엔드에 대한 추가 매개변수는 없습니다.	swift 기본값입니다.

2.2.1.2. NFS 백엔드

매개변수	설명	현재의
CinderBackupBackend	백업 리포지토리의 백엔드입니다.	nfs
CinderBackupNfsShare	백업을 저장하기 위해 마운트하려는 원격 NFS 공유입니다. 서버 이름 또는 IP 다음에 내보내기를 지정했는지 확인합니다.	
CinderBackupNfsMountOptions	선택 사항: NFS 공유를 마운트하기 위한 선택으로 구분된 옵션 목록입니다.	

2.2.1.3. Red Hat Ceph Storage 백엔드

매개변수	설명	현재의
CinderBackupBackend	백업 리포지토리의 백엔드입니다.	Ceph
CinderBackupRbdPoolName	백업을 저장하는 Ceph 클러스터의 RBD 풀 이름입니다.	백업

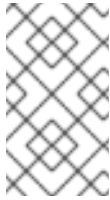
2.2.1.4. S3 백엔드

매개변수	설명	현재의
CinderBackupBackend	백업 리포지토리의 백엔드입니다.	s3
CinderBackupS3Bucket	백업을 저장하는 S3 버킷입니다. 참고 블록 스토리지 백업 서비스를 배포하기 전에 S3 백엔드에 이 버킷을 생성하고 이 버킷에 쓰는 데 필요한 권한을 구성했는지 확인합니다.	volumeBackups
CinderBackupS3AccessKey	S3 버킷에 연결하는 S3 액세스 키입니다.	
CinderBackupS3SecretKey	S3 버킷에 연결할 S3 시크릿 키입니다.	
CinderBackupS3EndpointUrl	S3 끝점의 URL입니다.	

2.2.2. 블록 스토리지 백업 서비스 구성

선택한 백엔드에서 지원하는 Block Storage 백업 서비스 매개변수를 구현할 수 있습니다.

매개변수	설명	현재의
------	----	-----

CinderBackupCompressionAlgorithm	백엔드에서 지원하는 경우 백업 리포지토리의 데이터 압축을 활성화할 수 있습니다. 데이터 압축에는 추가 CPU 전원이 필요하지만 네트워크 대역폭과 스토리지 공간을 줄일 수 있습니다.  참고 Red Hat Ceph Storage 백엔드 드라이버의 데이터 압축 알고리즘을 지정할 수 없습니다. 이 매개변수는 이 백엔드에 대해 무시됩니다.	zlib 대안: none,bzip2 또는 zstd
---	--	--

3장. 블록 스토리지 백업 서비스 사용

Block Storage 백업 서비스를 사용하여 전체 또는 증분 백업을 수행하고 백업을 보호하고 볼륨에 백업을 복원할 수 있습니다.

3.1. 백업 생성

볼륨이 실패하면 데이터가 손실되지 않도록 Block Storage 볼륨의 백업을 만듭니다. 자세한 내용은 [전체 볼륨 백업 생성](#) 을 참조하십시오. 볼륨의 스냅샷에서 직접 백업을 생성할 수도 있습니다. 자세한 내용은 [전체 스냅샷 백업 생성](#) 을 참조하십시오. 백업은 볼륨 데이터 외에도 이름 및 설명과 같은 볼륨 메타데이터를 저장합니다.

백업 리포지토리에 데이터 압축을 활성화한 경우 백업이 압축되므로 성능이 저하될 수 있습니다.

전체 백업은 더 쉽게 관리할 수 있지만 시간이 지남에 따라 볼륨의 크기가 증가할 때 리소스가 집약적일 수 있습니다. 증분 백업을 사용하면 볼륨에 대한 주기적인 변경 사항을 캡처하고 리소스 사용량을 최소화할 수 있습니다. 자세한 내용은 [증분 백업](#) 을 참조하십시오.

액세스할 수 있는 볼륨 백업을 생성할 수 있습니다. 프로젝트 관리자는 프로젝트에 속한 모든 볼륨을 백업할 수 있습니다. 이러한 백업은 관리자가 백업을 생성할 때 추가 인수를 제공하지 않는 한 볼륨 소유자에서 숨겨집니다. 자세한 내용은 [볼륨 소유자를 인증하는 백업 인수](#) 를 참조하십시오.

각 프로젝트(테넌트)는 최대 백업 수와 생성할 수 있는 모든 백업의 최대 총 크기를 제한합니다. 프로젝트 관리자는 이러한 할당량을 보고 변경할 수 있습니다. 자세한 내용은 [프로젝트 백업 할당량 보기 및 수정](#) 을 참조하십시오.

일반적으로 **사용 가능한** 상태의 볼륨만 백업할 수 있지만 필요한 경우 **사용 중인** 상태로 볼륨을 백업할 수 있습니다. 자세한 내용은 [사용 중인 볼륨의 백업 생성](#) 을 참조하십시오.

블록 스토리지 볼륨의 백업을 생성할 때 이 백업의 메타데이터는 이 볼륨을 복원할 때 사용되는 Block Storage 서비스 데이터베이스에 저장됩니다. 백업이 블록 스토리지 서비스 데이터베이스의 심각한 손실 유지되도록 프로젝트 관리자는 이 백업의 메타데이터를 수동으로 내보내고 저장할 수 있습니다. 자세한 내용은 [백업](#) 을 참조하십시오.

3.1.1. 전체 볼륨 백업 생성

볼륨 전체 백업을 하나 이상 생성할 수 있습니다.

사전 요구 사항

- 볼륨 소유자 및 프로젝트 관리자만 볼륨을 백업할 수 있습니다.
- 백업 리포지토리에 필요한 공간이 있어야 합니다.
- 프로젝트에 지정된 백업 할당량이 초과되지 않았습니. 자세한 내용은 [프로젝트 백업 할당량 보기 및 수정](#) 을 참조하십시오.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업할 볼륨의 ID 또는 이름을 가져올 볼륨을 나열합니다.

```
$ openstack volume list
```



참고

일반적으로 **사용 가능한** 상태의 볼륨만 백업할 수 있지만 필요한 경우 **사용 중인** 상태로 볼륨을 백업할 수 있습니다. 자세한 내용은 [사용 중인 볼륨의 백업 생성](#)을 참조하십시오.

4. 볼륨을 백업합니다.



참고

볼륨 소유자가 아닌 프로젝트 관리자인 경우 볼륨 소유자가 이 백업에 액세스할 수 있도록 하려면 이 백업을 생성할 때 추가 매개변수를 제공해야 합니다. 자세한 내용은 [볼륨 소유자를 인증하는 백업 인수](#)를 참조하십시오.

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

- **<volume>**을 백업하려는 볼륨의 ID 또는 이름으로 바꿉니다.
- 선택 사항: **<backup_name>**을 이 백업 이름으로 바꿉니다.
이 명령은 이 백업의 ID를 즉시 제공하지만 볼륨은 백그라운드에서 비동기적으로 백업됩니다.
예를 들면 다음과 같습니다.

```
$ openstack volume backup create --name vol1bu2 vol_1
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb |
| name  | vol1bu2 |
+-----+-----+
```

검증

- 백업을 나열합니다.

```
$ openstack volume backup list
```

이 백업에 **사용 가능한** 상태가 있으면 볼륨 백업이 생성됩니다. 예를 들면 다음과 같습니다.

```
+-----+-----+-----+-----+-----+
| ID | Name | Description | Status | Size |
+-----+-----+-----+-----+-----+
| 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None | available | 1 |
| b604a932-94a5-468e-bf6b-841ac16f69a8 | None | None | available | 1 |
+-----+-----+-----+-----+-----+
```

추가 리소스

- [백업 문제 해결](#)

- 백업 생성

3.1.2. 전체 스냅샷 백업 생성

스냅샷과 연결된 볼륨의 ID를 사용하여 스냅샷에서 전체 백업을 생성할 수 있습니다.

스냅샷에 직접 연결하여 백업이 생성됩니다. 이는 스냅샷을 볼륨에 복제한 다음 이 볼륨을 백업하는 것보다 빠릅니다. 그러나 이 기능은 스냅샷에서 볼륨을 생성하는 추가 단계 때문에 백업 성능에 영향을 미칠 수 있습니다.

사전 요구 사항

- 백업 리포지토리에 필요한 공간이 있어야 합니다.
- 프로젝트에 지정된 백업 할당량이 초과되지 않았습니니다. 자세한 내용은 [프로젝트 백업 할당량 보기 및 수정](#) 을 참조하십시오.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업할 스냅샷의 이름 또는 ID를 가져올 스냅샷을 나열합니다.

```
$ openstack volume snapshot list
```

4. 이 스냅샷과 연결된 볼륨의 ID를 가져오려면 이 스냅샷의 세부 정보를 나열합니다.

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

- **<snapshot>** 을 백업할 스냅샷의 이름 또는 ID로 바꿉니다.
volume_id 필드의 값은 이 스냅샷과 연결된 볼륨의 ID입니다. 예를 들면 다음과 같습니다.

```
$ 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. 스냅샷을 백업합니다.

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

- `<volume_id>`를 이 스냅샷과 연결된 볼륨의 ID로 바꿉니다.
- 선택 사항: `<backup_name>`을 이 백업 이름으로 바꿉니다.
이 명령은 이 백업의 ID를 즉시 제공하지만 스냅샷은 백그라운드에서 비동기식으로 백업됩니다. 예를 들면 다음과 같습니다.

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

검증

- 백업을 나열합니다.

```
$ openstack volume backup list
```

이 백업의 상태가 **사용 가능한** 경우 스냅샷 백업이 생성됩니다. 예를 들면 다음과 같습니다.

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

3.1.3. 사용 중인 볼륨의 백업 생성

일반적으로 **사용 가능한** 상태의 볼륨만 백업할 수 있습니다. 그러나 백업을 생성할 때 **--force** 옵션을 사용하여 사용 중 상태의 볼륨을 백업할 수 있습니다.

--force 볼륨 백업 옵션을 사용하는 경우 백업을 수행하기 전에 볼륨이 정지되지 않으므로 크래시 일관성을 생성하지만 애플리케이션 불일치는 생성하지 않습니다. 따라서 데이터는 그대로 유지되지만 백업은 백업이 수행되었을 때 실행 중인 애플리케이션에 대한 인식이 없습니다.

사전 요구 사항

- 볼륨 소유자 및 프로젝트 관리자만 볼륨을 백업할 수 있습니다.
- 백업 리포지토리에 필요한 공간이 있어야 합니다.
- 프로젝트에 지정된 백업 할당량이 초과되지 않았습니다. 자세한 내용은 [프로젝트 백업 할당량 보기 및 수정](#) 을 참조하십시오.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업할 볼륨의 ID 또는 이름을 가져올 볼륨을 나열합니다.

```
$ openstack volume list
```

예를 들면 다음과 같습니다.

```
+-----+-----+-----+-----+-----+
| 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. 백업할 볼륨이 **사용** 중이면 강제로 백업을 수행합니다.

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

- **<volume>**을 백업하려는 볼륨의 ID 또는 이름으로 바꿉니다.
- 선택 사항: **<backup_name>**을 이 백업 이름으로 바꿉니다.
이 명령은 이 백업의 ID를 즉시 제공하지만 볼륨은 백그라운드에서 비동기적으로 백업됩니다.
예를 들면 다음과 같습니다.

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

검증

- 백업을 나열합니다.

```
$ openstack volume backup list
```

이 백업에 **사용 가능한** 상태가 있으면 볼륨 백업이 생성됩니다. 예를 들면 다음과 같습니다.

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

추가 리소스

- [백업 문제 해결](#)

- 백업 생성

3.1.4. 증분 백업

볼륨에 하나 이상의 전체 백업이 있는 경우 Block Storage 백업 서비스를 사용하여 증분 백업을 생성할 수 있습니다. 자세한 내용은 [증분 백업 생성](#)을 참조하십시오.

전체 백업은 더 쉽게 관리할 수 있지만 시간이 지남에 따라 볼륨의 크기가 증가할 때 리소스가 집약적일 수 있습니다. 증분 백업을 사용하면 볼륨에 대한 주기적인 변경 사항을 캡처하고 리소스 사용량을 최소화할 수 있습니다.

증분 백업은 마지막 전체 또는 증분 백업 이후 볼륨에 대한 변경 사항만 저장합니다.

증분 백업은 백업 관리에 필요한 관리 오버헤드를 늘립니다. 예를 들어 이미 하나 이상의 증분 백업이 있는 경우 전체 백업을 삭제할 수 없으며 최신 증분 백업만 삭제할 수 있습니다. For example, you cannot delete a full backup if it already has one or more incremental backups, you can only delete the latest incremental backup.

증분 백업은 전체 백업보다 성능이 저하됩니다. 증분 백업을 만들 때 볼륨의 모든 데이터를 먼저 읽고 전체 백업 및 후속 증분 백업의 데이터와 비교해야 합니다.

3.1.4.1. 증분 백업 생성

증분 백업을 생성하여 마지막 전체 또는 증분 백업 이후 볼륨에 대한 변경 사항만 저장할 수 있습니다.

사전 요구 사항

- 볼륨 전체 백업이 하나 이상 있어야 합니다. 자세한 내용은 [전체 볼륨 백업 생성](#)을 참조하십시오.
- 볼륨 소유자 및 프로젝트 관리자만 볼륨을 백업할 수 있습니다.
- 백업 리포지토리에 필요한 공간이 있어야 합니다.
- 프로젝트에 지정된 백업 할당량이 초과되지 않았습니. 자세한 내용은 [프로젝트 백업 할당량 보기 및 수정](#)을 참조하십시오.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업할 볼륨의 ID 또는 이름을 가져올 볼륨을 나열합니다.

```
$ openstack volume list
```

4. 볼륨을 백업하고 **--incremental** 옵션을 사용합니다.

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

- **<volume>**을 백업하려는 볼륨의 ID 또는 이름으로 바꿉니다.
- 선택 사항: **<backup_name>**을 이 백업 이름으로 바꿉니다.

이 명령은 이 백업의 ID를 즉시 제공하지만 볼륨은 백그라운드에서 비동기적으로 백업됩니다. 예를 들면 다음과 같습니다.

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

검증

- 백업을 나열합니다.

```
$ openstack volume backup list
```

이 백업에 **사용 가능한** 상태가 있으면 볼륨 백업이 생성됩니다. 예를 들면 다음과 같습니다.

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

추가 리소스

- [백업 문제 해결](#)

3.1.5. 백업 성능 고려 사항

중분 백업 및 데이터 압축과 같은 블록 스토리지 백업 서비스의 일부 기능은 백업의 성능을 줄일 수 있습니다.

볼륨에 대한 주기적인 변경 사항만 캡처하면 중분 백업으로 리소스 사용량을 최소화할 수 있습니다. 자세한 내용은 [중분 백업](#)을 참조하십시오. 그러나 중분 백업은 전체 백업보다 성능이 낮습니다. 중분 백업을 만들 때 볼륨의 모든 데이터를 먼저 읽고 전체 백업 및 후속 중분 백업의 데이터와 비교해야 합니다.

스냅샷에 직접 연결하여 백업을 생성할 수도 있습니다. 이는 스냅샷을 볼륨에 복제하는 것보다 빠릅니다. 자세한 내용은 [전체 스냅샷 백업 생성](#)을 참조하십시오. 그러나 이 기능은 스냅샷에서 볼륨을 생성하는 추가 단계 때문에 백업 성능에 영향을 미칠 수 있습니다.

백업 리포지토리의 데이터 압축을 활성화하려면 추가 CPU 성능이 필요하지만 네트워크 대역폭과 스토리지 공간을 전반적으로 덜 사용합니다. 백업 리포지토리의 데이터 압축을 활성화하거나 비활성화하도록 Block Storage 백업 서비스를 구성할 수 있습니다. 자세한 내용은 [Block Storage 백업 서비스 구성](#)을 참조하십시오.



참고

Red Hat Ceph Storage 백엔드에서는 데이터 압축을 사용할 수 없습니다.

3.1.6. 볼륨 소유자를 인증하는 백업 인수

프로젝트 관리자는 프로젝트에 속한 모든 볼륨을 백업할 수 있지만 이러한 백업은 볼륨 소유자에서 숨겨집니다.

볼륨 소유자가 볼륨 백업에도 액세스할 수 있도록 프로젝트 관리자는 볼륨을 백업할 때 볼륨 소유자로 인증하기 위해 다음과 같은 추가 인수를 제공해야 합니다.

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

- **<keystoneurl>**을 ID 서비스의 URL 끝점으로 바꿉니다. 일반적으로 <http://IP:5000/v3>. 여기서 **IP**는 ID 서비스 호스트의 IP 주소입니다.
- **<projectname>**을 볼륨 소유자의 프로젝트(테넌트) 이름으로 바꿉니다.
- **<username>** 및 **<password>**를 이 프로젝트 내의 볼륨 소유자인 사용자의 사용자 이름 및 암호 자격 증명으로 바꿉니다.



참고

[--name <backup_name>] <volume>은 볼륨 백업을 생성할 때 일반적인 인수입니다.

3.1.7. 프로젝트 백업 할당량 보기 및 수정

프로젝트 관리자는 백업의 최대 수와 특정 프로젝트(테넌트)에 대해 생성할 수 있는 모든 백업의 최대 총 크기(**GB**)를 변경하거나 볼 수 있으며 이 프로젝트의 백업 할당량 사용을 확인할 수 있습니다.

사전 요구 사항

- 프로젝트의 백업 할당량을 보거나 수정하려면 프로젝트 관리자여야 합니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 필요한 프로젝트의 **ID** 또는 이름을 가져올 프로젝트를 나열합니다.

```
$ openstack project list
```

4.

특정 프로젝트의 백업 할당량을 확인합니다.

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

•

<project>를 필요한 프로젝트의 ID 또는 이름으로 바꿉니다.

테이블의 **backup-gigabytes** 필드 값은 이 프로젝트에서 생성할 수 있는 모든 백업의 최대 총 크기입니다. 표의 **backups** 필드 값은 이 프로젝트에서 생성할 수 있는 최대 백업 수입니다. 예를 들면 다음과 같습니다.

+

```
$ openstack quota show c2c1da89ed1648fc8b4f35a045f8d34c
```

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

5.

프로젝트에 대해 생성된 모든 백업의 최대 총 크기를 수정합니다.

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

•

이 프로젝트에 대해 생성할 수 있는 백업의 최대 총 크기(GB)로 < maxgb >를 바꿉니다.

6.

프로젝트에 대해 생성할 수 있는 최대 백업 수를 수정합니다.

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

•

< maxnum >을 이 프로젝트에 생성할 수 있는 최대 백업 수로 바꿉니다.

7.

특정 프로젝트에 대한 이러한 백업 할당량의 사용량을 확인합니다.

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

<project_id>를 프로젝트 ID로 바꿉니다.

예를 들면 다음과 같습니다.

```
$ 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    |           |
+-----+-----+-----+-----+
```

검증

다음 할당량 중 하나를 변경한 경우 다음 변경 사항을 검토하십시오.

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

수정된 값이 테이블의 **backup-gigabytes** 및 **backups** 필드에 의해 지정되었는지 확인합니다. 예를 들면 다음과 같습니다.

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

3.1.8. 백업 취소

백업에 강제 삭제를 요청하여 취소해야 합니다.



중요

백업 리포지토리에 **Red Hat Ceph Storage** 백엔드를 사용하는 경우 백업을 취소할 수 없습니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 취소할 백업의 **ID** 또는 이름을 가져올 백업을 나열합니다.

```
$ openstack volume backup list
```

4. 백업을 취소합니다.

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

- **<backup>**을 취소하려는 볼륨 백업의 **ID** 또는 이름으로 바꿉니다.

백업이 성공적으로 취소되기까지 약간의 지연이 있을 수 있습니다.

검증

- 이 명령으로 백업 레코드가 나열되지 않으면 백업이 취소됩니다.

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

3.2. 에지 사이트 간 백업 및 복원

에지 사이트 및 가용성 영역의 **DCN(Distributed Compute Node)** 아키텍처에서 **Block Storage** 서비스(**cinder**) 볼륨을 백업하고 복원할 수 있습니다. **cinder-backup** 서비스는 중앙 가용성 영역(**AZ**)에서 실행되며 백업은 중앙 **AZ**에 저장됩니다. 블록 스토리지 서비스는 **DCN** 사이트에 백업을 저장하지 않습니다.

사전 요구 사항

- 선택적 **Block Storage** 백업 서비스를 배포합니다. 자세한 내용은 [블록 스토리지 볼륨 백업의 블록 스토리지 백업 서비스 배포](#)를 참조하십시오.
- **Block Storage(cinder) REST API** 마이크로 버전 **3.51** 이상
- 모든 사이트에서 공통 **openstack cephx** 클라이언트 이름을 사용해야 합니다. 자세한 내용은 **DCN(Distributed Compute Node) 아키텍처 배포**의 [외부 액세스용 Ceph 키 생성](#)을 참조하십시오.

프로세스

1. 첫 번째 **DCN** 사이트에 볼륨 백업을 생성합니다.

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

- **<volume_backup>**을 볼륨 백업의 이름으로 바꿉니다.
- **<az_central>**을 **cinder-backup** 서비스를 호스팅하는 중앙 가용성 영역의 이름으로 바꿉니다.
- **<edge_volume>**을 백업할 볼륨의 이름으로 바꿉니다.



참고

Ceph 인증 키에 문제가 있는 경우 호스트에서 컨테이너로 인증 키를 복사하도록 **cinder-backup** 컨테이너를 다시 시작해야 할 수 있습니다.

2.

두 번째 **DCN** 사이트의 새 볼륨으로 백업을 복원합니다.

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

- **<az_2>**를 백업을 복원하려는 가용성 영역의 이름으로 바꿉니다.
- **<new_volume>**을 새 볼륨의 이름으로 바꿉니다.
- **<volume_backup>**을 이전 단계에서 생성한 볼륨 백업 이름으로 바꿉니다.
- **<volume_size>**를 원래 볼륨의 크기와 같거나 큰 값(**GB**)으로 바꿉니다.

3.3. 백업 보호

블록 스토리지 볼륨의 백업을 생성할 때 이 백업의 메타데이터는 이 볼륨을 복원하는 데 사용되는 블록 스토리지 서비스 데이터베이스에 저장됩니다. 백업이 블록 스토리지 서비스 데이터베이스의 심각한 손실에 유지되도록 프로젝트 관리자는 이 백업의 메타데이터를 오프사이트 백업과 같은 안전한 위치에 수동으로 내보내고 저장할 수 있습니다. 자세한 내용은 [백업 메타데이터 내보내기](#)를 참조하십시오.

블록 스토리지 서비스 데이터베이스에 심각한 손실이 발생하면 이 데이터베이스에 백업을 복원할 때 사용되는 백업 메타데이터가 포함되어 있기 때문에 백업을 복원할 수 없습니다. 그러나 프로젝트 관리자가 수동으로 내보내기하고 백업의 메타데이터를 저장한 경우 프로젝트 관리자는 이 메타데이터를 새 블록 스토리지 데이터베이스로 가져올 수 있으므로 이 백업을 사용하여 볼륨을 복원할 수 있습니다. 자세한 내용은 [백업 메타데이터 가져오기](#)를 참조하십시오.



참고

중분 백업의 경우 이전 백업의 메타데이터를 가져와 볼륨을 복원할 수 있습니다.

3.3.1. 백업 메타데이터 내보내기

프로젝트 관리자는 백업의 메타데이터를 내보내 파일에 저장할 수 있으므로 블록 스토리지 데이터베이스에 치명적인 손실이 발생하더라도 볼륨 백업을 복원할 수 있습니다. 자세한 내용은 [백업](#)을 참조하십시오.



참고

중분 백업의 경우 이전 백업의 메타데이터를 내보내야 합니다.

사전 요구 사항

- 백업 메타데이터를 내보내려면 프로젝트 관리자여야 합니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업을 나열하여 백업의 **ID** 또는 이름을 가져옵니다.

```
$ openstack volume backup list
```

4. 백업의 메타데이터를 내보내 적절하게 이름이 지정된 **YAML** 파일에 저장합니다.

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

- **<backup>**을 볼륨 백업의 **ID** 또는 이름으로 바꿉니다.
- 이 백업에 대해 내보낸 **backup_service** 및 **backup_url** 값을 저장하려면 **<filename>**을 **YAML** 파일의 이름으로 바꿉니다.

예를 들면 다음과 같습니다.

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

5. 파일을 오프사이트 백업과 같은 안전한 위치에 복사합니다.

검증

- 파일을 편집하여 **backup_service** 및 **backup_url**의 값이 이 명령에서 제공하는 값과 일치하는지 확인합니다.

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

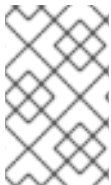
예를 들면 다음과 같습니다.

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

3.3.2. 백업 메타데이터 가져오기

프로젝트 관리자가 볼륨 백업의 메타데이터를 내보내고 저장한 경우 블록 스토리지 서비스 데이터베이스의 치명적인 손실 후 프로젝트 관리자가 이 메타데이터를 가져올 수 있습니다.

이 절차를 사용하여 삭제된 백업을 다시 생성할 수도 있습니다.



참고

중분 백업의 경우 이전 백업의 메타데이터도 가져와야 합니다.

사전 요구 사항

- 볼륨 백업 메타데이터를 블록 스토리지 데이터베이스로 가져오려면 프로젝트 관리자여야 합니다.
- 이 백업의 **backup_service** 및 **backup_url** 메타데이터 값을 제공해야 합니다. 자세한 내용은 [백업 메타데이터 내보내기](#)를 참조하십시오.
- 이 백업이 아직 포함되지 않은 블록 스토리지 데이터베이스입니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 이 백업의 내보낸 **backup_service** 및 **backup_url** 메타데이터 값을 저장한 파일을 찾습니다.

4. 이 블록 백업의 메타데이터 값을 **Block Storage** 데이터베이스로 가져옵니다.

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

- **<backup_service>**를 이 블록 백업의 **backup_service** 메타데이터 값으로 바꿉니다.

- **<backup_url>**을 이 블록 백업의 **backup_url** 메타데이터 값으로 바꿉니다.

이 명령은 이 백업의 이름과 ID를 제공합니다. 예를 들면 다음과 같습니다.

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

다음 단계

- [백업 복원](#)

3.4. 백업 복원

블록 스토리지 블록 백업을 생성한 후 필요한 경우 이 백업 데이터를 복원할 수 있습니다.

다음 방법 중 하나를 사용하여 백업을 복원할 수 있습니다.

- 지정한 볼륨에 백업을 복원합니다. 자세한 내용은 [특정 볼륨에 대한 백업 복원](#)을 참조하십시오.
- 백업을 새 볼륨으로 복원합니다. 자세한 내용은 [새 볼륨에 백업 복원](#)을 참조하십시오.



중요

블록 스토리지 서비스 데이터베이스에 치명적인 손실이 발생하면 메타데이터를 내보내고 저장한 경우를 제외하고 백업을 복원할 수 없습니다. 자세한 내용은 [백업](#)을 참조하십시오.

프로젝트 관리자만 볼륨 백업 복원을 취소할 수 있습니다. 자세한 내용은 [백업 복원 취소](#)를 참조하십시오.

3.4.1. 특정 볼륨으로 백업 복원

볼륨 백업을 이미 생성한 사용 가능한 볼륨에 복원할 수 있습니다.

암호화된 백업에서 볼륨을 복원하는 경우 대상 볼륨 유형도 암호화해야 합니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.
2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

\$ source ~/stackrc
3. 복원할 백업의 이름 또는 **ID**를 가져오도록 백업을 나열합니다.

```
$ openstack volume backup list
```

예를 들면 다음과 같습니다.

```
+-----+-----+-----+-----+
| ID           | Name   | Description | Status  | Size |
+-----+-----+-----+-----+
| 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None       | available | 1 |
```

4.

볼륨을 나열합니다.

```
$ openstack volume list
```

필요한 볼륨의 상태가 사용 가능한지 확인한 다음 이 볼륨의 이름 또는 ID를 가져옵니다. 예를 들면 다음과 같습니다.

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

5.

백업을 볼륨으로 복원합니다.

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

- **<backup>**을 **Block Storage** 볼륨 백업의 이름 또는 ID로 바꿉니다.
- **<volume>**을 사용 가능한 블록 스토리지 볼륨의 이름 또는 ID로 바꿉니다.

예를 들면 다음과 같습니다.

```
$ openstack volume backup restore vol1bu2 vol_3
+-----+-----+
| Field   | Value                               |
+-----+-----+
| backup_id | 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb |
| volume_id | 654e2be8-bc79-4528-96a7-5f773d31c201 |
| volume_name | vol_3                               |
+-----+-----+
```

6.

이 명령에서 제공하는 **backup_id**가 복원된 백업의 ID와 일치하고 **volume_name** 및

volume_id 값이 지정된 볼륨의 이름 및 ID에 해당하는지 확인합니다.

7. 더 이상 필요하지 않은 경우 백업을 삭제합니다.

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

3.4.2. 새 볼륨으로 백업 복원

블록 스토리지 볼륨의 백업을 복원할 때 새 볼륨을 생성할 수 있습니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 복원할 백업의 이름 또는 ID를 가져오도록 백업을 나열합니다.

```
$ openstack volume backup list
```

예를 들면 다음과 같습니다.

```
+-----+-----+-----+-----+-----+
| ID              | Name   | Description | Status  | Size |
+-----+-----+-----+-----+-----+
| 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb | vol1bu2 | None       | available | 1 |
```

4. 백업을 새 볼륨으로 복원합니다.

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

- **<backup>**을 **Block Storage** 볼륨 백업의 이름 또는 ID로 바꿉니다.

예를 들면 다음과 같습니다.

```
$ cinder backup-restore vol1bu2
+-----+
| Property | Value |
+-----+
| backup_id | 83dadc43-2aa9-4c0b-bc05-a12203a8f4cb |
| volume_id | 296c853c-c749-4eb6-857a-57ec182232a6 |
| volume_name | restore_backup_83dadc43-2aa9-4c0b-bc05-a12203a8f4cb |
+-----+
```

5.

이 명령에서 제공하는 **backup_id** 가 복원된 백업의 ID에 해당하는지 확인합니다.

volume_id 값은 생성된 볼륨의 ID입니다. 그러나 **volume_name** 은 백업된 볼륨의 이름으로 교체되는 임시 이름일 수 있습니다.

6.

볼륨을 나열하여 ID가 **volume_id** 인 볼륨이 생성되었는지 확인하고 이 볼륨 이름을 가져옵니다.

```
$ openstack volume list
```

예를 들면 다음과 같습니다.

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

7.

더 이상 필요하지 않은 경우 백업을 삭제합니다.

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

3.4.3. 백업 복원 취소

프로젝트 관리자는 백업 상태를 오류로 변경하여 볼륨 백업 복원을 취소할 수 있습니다. 하지만 **Red Hat Ceph Storage**가 백업 리포지토리의 백엔드인 경우 백업 복원을 취소할 수 없습니다.



주의

백업을 시작한 후 복원을 취소하면 실제로 복원된 데이터 양을 알 수 있는 방법이 없기 때문에 대상 볼륨은 쓸모 없습니다.

사전 요구 사항

- 볼륨 백업 복원을 취소하려면 프로젝트 관리자여야 합니다.
- 백업 리포지토리의 백엔드가 **Red Hat Ceph Storage**가 아닌지 확인합니다.

프로세스

1. 언더클라우드 호스트에 **stack** 사용자로 로그인합니다.

2. **stackrc** 언더클라우드 인증 정보 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 백업을 나열하여 복원을 중지하려는 백업의 이름 또는 ID를 가져옵니다.

```
$ openstack volume backup list
```

4. 복원 작업을 취소하려면 이 백업의 상태를 **error** 로 변경합니다.

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

- **<backup>** 을 복원하지 않으려는 볼륨 백업의 이름 또는 ID로 바꿉니다.

백업 리포지토리의 백엔드에서 복원을 취소하기 전에 백업 상태의 변경 사항을 감지해야 하므로 복원을 취소하는 것은 비동기 작업입니다.

검증

- 볼륨 백업을 나열하여 복원이 취소되었는지 확인합니다.

```
$ openstack volume backup list
```

백업 상태가 **available** 로 변경되면 복원이 취소됩니다.

4장. 블록 스토리지 백업 서비스 문제 해결

블록 스토리지 서비스가 올바르게 실행되고 있는지 확인한 다음 로그 파일에 오류 메시지가 있는지 확인하여 많은 문제를 진단할 수 있습니다.

4.1. 블록 스토리지 백업 서비스 배포 확인

배포 후 또는 문제를 해결할 때 필요한 블록 스토리지 서비스가 호스트에서 올바르게 실행되고 있는지 확인하는 것이 중요합니다. 블록 스토리지 스케줄러 서비스와 같이 **Block Storage** 백업 서비스가 모든 컨트롤러 노드에서 실행되고 있는지 확인합니다.

필요한 블록 스토리지 서비스가 올바르게 실행되고 있는지 확인한 후 블록 스토리지 백업 서비스가 성공적으로 배포되었는지 확인해야 합니다.

프로세스

1.

openstack volume service list 명령을 실행합니다.

```
# 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.

모든 서비스의 상태 항목이 **up** 인지 확인합니다. 그렇지 않은 경우 관련 로그 파일을 검사합니다. 이러한 로그 파일의 위치에 대한 자세한 내용은 [오버클라우드 관찰 기능 관리의 Block Storage\(cinder\) 로그](#) 파일을 참조하십시오.

3.

블록 스토리지 볼륨을 백업하고 백업에 성공했는지 확인하여 블록 스토리지 백업 서비스가 성공적으로 배포되었는지 확인합니다. 자세한 내용은 [백업 문제 해결](#)을 참조하십시오.

수거 디스크

- 블록 스토리지 백업 서비스 로그 파일 검사

4.2. 백업 문제 해결

블록 스토리지 백업 서비스는 블록 스토리지(**cinder**) 볼륨을 백업하라는 요청을 받을 때 정적 검사를 수행합니다. 이러한 검사가 실패하면 즉시 알람을 받습니다.

- 잘못된 볼륨 참조(이/가누락됨)를 확인합니다.
- 볼륨이 사용 중이거나 인스턴스에 연결되어 있는지 확인합니다. 사용 중인 경우 **--force** 옵션을 사용하여 백업을 수행해야 합니다. 자세한 내용은 [사용 중인 볼륨의 백업 생성](#)을 참조하십시오.

--force 볼륨 백업 옵션을 사용하는 경우 백업을 수행하기 전에 볼륨이 정지되지 않으므로 크래시 일관성을 생성하지만 애플리케이션 불일치는 생성하지 않습니다. 따라서 데이터는 그대로 유지되지만 백업은 백업이 수행되었을 때 실행 중인 애플리케이션에 대한 인식이 없습니다.

이러한 검사가 성공하면 **Block Storage** 백업 서비스에서 이 볼륨에 대한 요청을 수락하고 **CLI** 백업 명령이 즉시 반환되고 볼륨이 백그라운드에서 백업됩니다.

따라서 **CLI backup** 명령은 백업이 실패하더라도 반환됩니다. **openstack volume backup list** 명령을 사용하여 백업 항목의 **Status** 를 사용할 수 있을 때 볼륨 백업에 성공했는지 확인할 수 있습니다.

백업에 실패하면 오류 메시지가 있는지 **Block Storage** 백업 서비스 로그 파일을 검사하여 원인을 검색합니다. 자세한 내용은 [블록 스토리지 백업 서비스 로그 파일](#) 테스트를 참조하십시오.

4.3. 블록 스토리지 백업 서비스 로그 파일 검사

백업 또는 복원에 성공하지 못하면 이유를 결정하는 데 도움이 될 수 있는 오류 메시지에 대해 **Block Storage** 백업 서비스 로그 파일을 검사할 수 있습니다.

프로세스

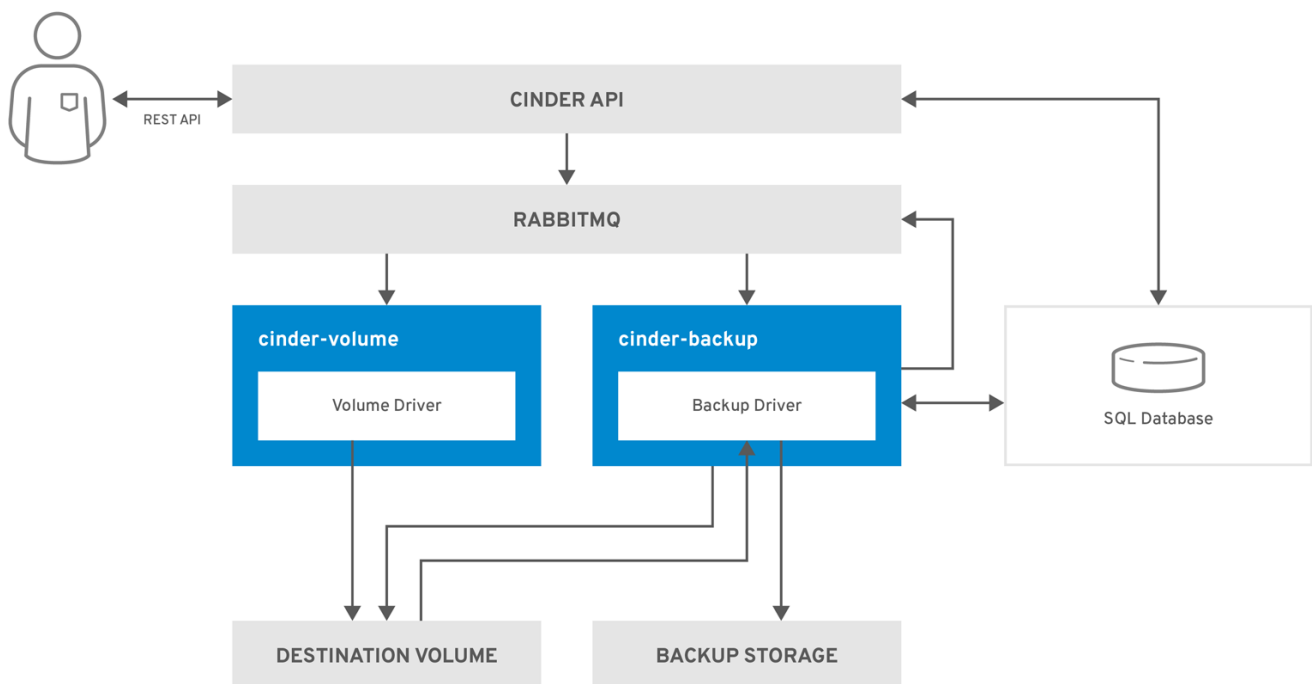
- 백업 서비스가 실행 중인 컨트롤러 노드에서 블록 스토리지 백업 서비스 로그 파일을 찾습니다.

이 로그 파일은 `/var/log/containers/cinder/cinder-backup.log` 경로에 있습니다.

4.4. 볼륨 백업 워크플로

다음 다이어그램 및 설명은 사용자가 **cinder API**를 **Block Storage(cinder)** 볼륨을 백업하도록 요청할 때 발생하는 단계를 설명합니다.

그림 4.1. 블록 스토리지 볼륨의 백업 생성



OPENSTACK_483337_1218

1. 사용자는 **REST API**인 **cinder API**에 대한 요청을 발행하여 블록 스토리지 볼륨을 백업합니다.
2. **cinder API**는 **HAProxy**에서 요청을 수신하고 요청, 사용자 자격 증명 및 기타 정보를 검증합니다.
3. **cinder API**는 **SQL** 데이터베이스에 백업 레코드를 생성합니다.
4. **cinder API**는 **AMQP**를 통해 **cinder-backup** 서비스를 비동기적으로 호출하여 볼륨을 백업합니다.

5.
cinder API는 ID가 있는 현재 백업 레코드를 **API** 호출자에게 반환합니다.
6.
RPC 생성 메시지가 백업 서비스 중 하나에 도착합니다.
7.
cinder-backup 서비스는 **get_backup_device**에 동기 **RPC** 호출을 수행합니다.
8.
cinder-volume 서비스를 통해 올바른 장치가 호출자로 반환됩니다. 일반적으로 동일한 볼륨이지만 볼륨이 사용 중인 경우 구성에 따라 서비스는 임시 복제 볼륨 또는 임시 스냅샷을 반환합니다.
9.
cinder-backup 서비스는 다른 동기 **RPC**를 **cinder-volume**으로 발행하여 소스 장치를 노출합니다.
10.
cinder-volume 서비스는 소스 장치(**volume** 또는 **snapshot**)를 내보내고 매핑하고 적절한 연결 정보를 반환합니다.
11.
cinder-backup 서비스는 연결 정보를 사용하여 소스 장치를 연결합니다.
12.
cinder-backup 서비스는 장치가 이미 연결된 백업 백엔드 드라이버를 호출하여 백업 리포지토리로 데이터 전송을 시작합니다.
13.
소스 장치는 백업 호스트에서 분리됩니다.
14.
cinder-backup 서비스는 동기 **RPC**를 **cinder-volume**으로 발행하여 소스 장치의 연결을 해제합니다.
15.
cinder-volume 서비스는 매핑 해제 및 장치의 내보내기를 제거합니다.
16.
임시 볼륨 또는 임시 스냅샷을 만든 경우 **cinder-backup**은 **cinder-volume**을 호출하여 제거합니다.

17.

cinder-volume 서비스는 임시 볼륨을 제거합니다.

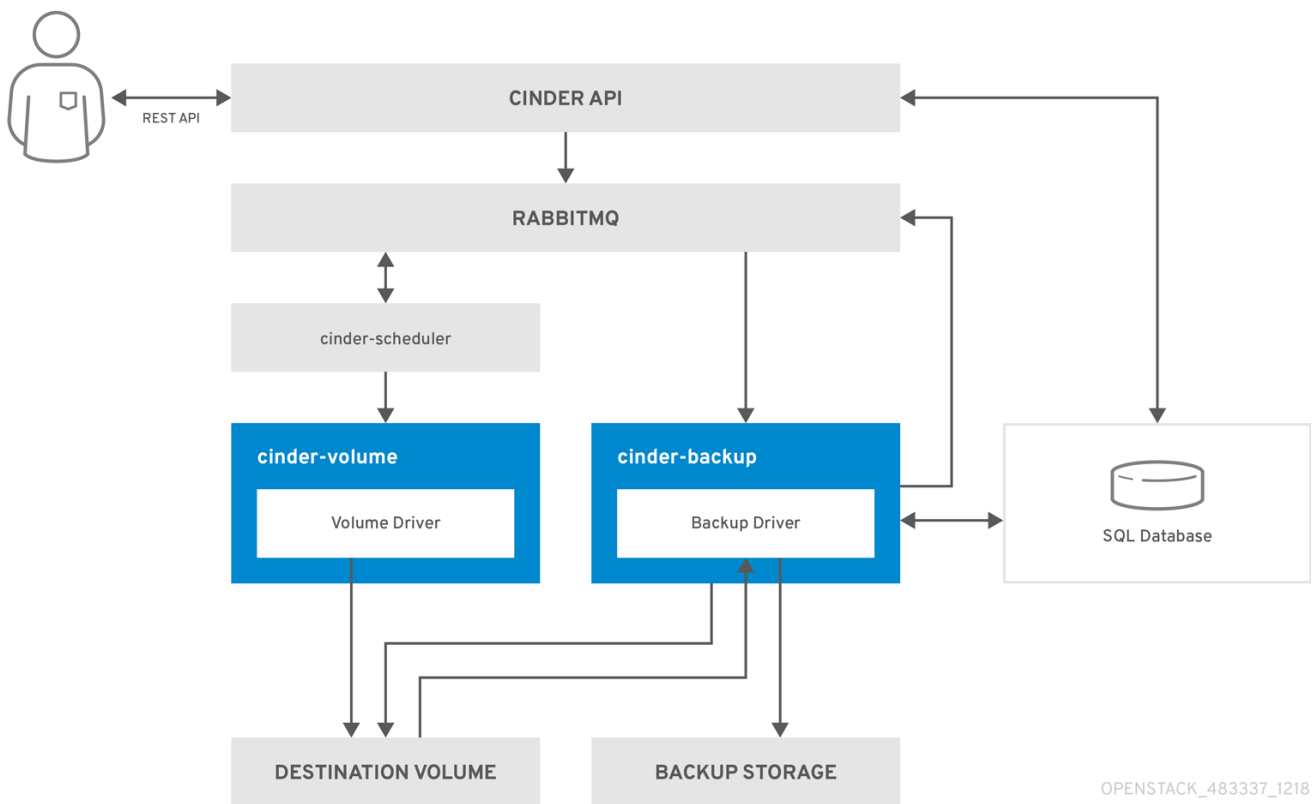
18.

백업이 완료되면 백업 레코드가 데이터베이스에서 업데이트됩니다.

4.5. 볼륨 복원 워크플로

다음 다이어그램 및 설명은 사용자가 **cinder API**를 요청하여 **Block Storage 서비스(cinder)** 백업을 복원할 때 발생하는 단계를 설명합니다.

그림 4.2. 블록 스토리지 백업 복원



1.

사용자는 블록 스토리지 백업을 복원하기 위해 **REST API**인 **cinder API**에 대한 요청을 발행합니다.

2.

cinder API는 **HAProxy**에서 요청을 수신하고 요청, 사용자 자격 증명 및 기타 정보를 검증합니다.

3.

요청에 기존 볼륨이 대상으로 포함되지 않은 경우 **cinder API**는 비동기 **RPC**를 호출하여 새 볼륨을 만들고 사용 가능하게 될 때까지 볼륨의 상태를 폴링합니다.

4. **cinder-scheduler** 에서 볼륨 서비스를 선택하고 **RPC**를 호출하여 볼륨을 생성합니다.
5. 선택한 **cinder-volume** 서비스는 볼륨을 생성합니다.
6. **cinder API**에서 볼륨을 사용할 수 있음을 감지하면 백업 레코드가 데이터베이스에 생성됩니다.
7. **cinder API**는 **AMQP**를 통해 비동기식 **RPC**를 호출하여 백업을 복원합니다.
8. **cinder API**는 현재 볼륨 ID, 백업 ID 및 볼륨 이름을 **API** 호출자로 반환합니다.
9. **RPC** 생성 메시지가 백업 서비스 중 하나에 도착합니다.
10. **cinder-backup** 서비스는 **cinder-volume** 에 대한 동기 **RPC** 호출을 수행하여 볼륨을 노출합니다.
11. **cinder-volume** 서비스는 적절한 연결 정보를 반환하는 볼륨을 내보내고 매핑합니다.
12. **cinder-backup** 서비스는 연결 정보를 사용하여 볼륨을 연결합니다.
13. **cinder-backup** 서비스는 이미 연결된 볼륨이 있는 백엔드 드라이버를 호출하므로 볼륨으로 데이터 복원이 시작됩니다.
14. 볼륨이 백업 호스트에서 분리됩니다.
15. **cinder-backup** 서비스는 동기 **RPC**를 **cinder-volume** 으로 발행하여 볼륨 연결을 해제합니다.
16. **cinder-volume** 서비스는 매핑 해제 및 볼륨의 내보내기를 제거합니다.

17.

볼륨이 복원되면 데이터베이스에서 백업 레코드가 업데이트됩니다.