4.7 블록 스토리지 서비스

OpenStack 블록 스토리지 서비스는 게스트 인스턴스에 블록 스토리지 디바이스를 제공하는 서비스이다. 이 서비스에서는 블록 스토리지 드라이버나 다양한 벡엔드 구성을 통해 스토리지가 제공되고 사용된다. 지원되는 블록스토리지 드라이버로는 NAS/SAN, NFS, iSCSI, Chep 등이 있으며 블록 스토리지 API 와 스케쥴러는 일반적으로 컨트롤러 노드에서 실행되고 볼륨 서비스는 사용되는 드라이버에 따라 컨트롤러 노드나 컴퓨트 노드 또는 별도의 블록 스토리지 노드에서 실행된다. 더 자세한 정보는 설정 레퍼런스를 살펴보기 바란다.

4.7.1 블록 스토리지 서비스 개요

OpenStack 블록 스토리지 서비스(cinder)에서는 가상 머신에 고정적인 스토리지를 추가할 수 있다. 블록 스토리지는 볼륨 관리를 위한 인프라와 함께 OpenStack 컴퓨트와 상호작용하여 인스턴스에 볼륨을 제공한다. 이와 함께 볼륨 스냅샷 및 볼륨 유형 관리가 가능하다.
블록 스토리지 서비스는 다음의 구성요소로 이루어져 있다.

- cinder-api
  API 요청을 받아들이고 이를 cinder-volume 에 전달하여 동작하도록 한다.
- cinder-volume
  블록 스토리지 서비스와 직접 상호작용하며 cinder-scheduler 와 같은 프로세스와 메시지 큐를 통해 상호작용한다. cinder-volume 서비스는 블록 스토리지 서비스로 전송되는 읽기 쓰기 요청을 처리하여 상태를 유지한다. 다양한 드라이버 아키텍처를 통해 다양한 스토리지 유형에서 동작한다.
- cinder-scheduler 데몬
  nova-scheduler 와 유사하게 볼륨 생성을 위해 최적의 스토리지 제공 노드를 선택한다.
- cinder-backup 데몬
  백업 스토리지 장비에 모든 유형의 볼륨 백업을 제공하는 서비스로 cinder-volume 서비스와 마찬가지로 다양한 드라이버 아키텍처를 통해 다양한 스토리지 유형에서 동작한다.
- 메시징 큐
  블록 스토리지 프로세스 간에 정보를 전달한다.

4.7.2 컨트롤러 노드 설치 및 구성

이 섹션에서는 컨트롤러 노드에 블록 스토리지 서비스인 cinder 를 설치하고 구성하는 과정에 대해서 설명한다. 블록 스토리지 서비스에는 인스턴스에 볼륨을 제공하기 위한 추가적인 스토리지 노드가 필요한데, 이 가이드에서는 컨트롤러 노드를 블록스토리지 노드로 사용한다.

● 사전 준비 사항

OpenStack 블록 스토리지(cinder) 서비스 설치와 구성에 앞서 데이터베이스, 서비스 인증 및 API 엔드포인트 생성을 수행한다.

1. 데이터베이스는 다음의 과정을 통해 생성한다.

a. 데이터베이스 접속 클라이언트를 사용하여 데이터베이스 서버에 root 로 로그인한다. 이 때 암호는 앞에서 설정한 *dbpassword* 이다.

```
# mysql -u root -p
Enter password:
```

b. cinder 데이터베이스를 생성한다.

```
MariaDB [(none)]> CREATE DATABASE cinder;
```

c. cinder 데이터베이스에 권한을 부여한다. 데이터베이스 암호는 앞에서 설정한 *cinderdbpass* 를 사용한다. 모든 명령은 줄 바꿈 없이 한 줄로 입력한다.

```
MariaDB [(none)]> GRANT ALL PRIVILEGES on cinder.* TO 'cinder'@'localhost'
IDENTIFIED BY 'cinderdbpass';
MariaDB [(none)]> GRANT ALL PRIVILEGES on cinder.* TO 'cinder'@'%' IDENTIFIED BY
'cinderdbpass';
```

d. 데이터베이스 클라이언트를 종료한다.

2. 관리자 모드로 CLI 명령을 실행하기 위해 admin 인증 스크립트를 실행한다.

```
# source /root/admin-openrc
```

3. 다음의 과정으로 서비스 인증을 생성한다.

a. cinder 사용자를 생성한다. 암호는 *cinderpass* 를 사용한다.

```
# openstack user create --domain default --password-prompt cinder
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | dfff203e4310474ab2ff3f8de8c63a98 |
| enabled   | True                             |
| id        | 0ef21b0eccb545ae81e3834f788861ed |
| name      | cinder                           |
+-----------+----------------------------------+
```

b. cinder 사용자에 admin 역할을 부여한다.(이 명령은 출력이 나타나지 않는다)

```
# openstack role add --project service --user cinder admin
```

c. cinder 와 cinderv2 서비스 엔티티를 생성한다. 블록 스토리지 서비스에는 두 개의 서비스 엔티티가 필요하다.

```
# openstack service create --name cinder --description "OpenStack Block Storage"
volume
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Block Storage          |
| enabled     | True                             |
| id          | 7db5c4ba9044469b837cd95d2523e41f |
| name        | cinder                           |
| type        | volume                           |
+-------------+----------------------------------+
# openstack service create --name cinderv2 --description "OpenStack Block Storage"
volumev2
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Block Storage          |
| enabled     | True                             |
| id          | 2e8f964acce84db7bb846df4e235942e |
| name        | cinderv2                         |
| type        | volumev2                         |
+-------------+----------------------------------+
```

4. cinder 와 cinderv2 에 대해서 각각 public, internal, admin 세 가지의 서비스 API 엔드포인트를 생성한다. 모든 명령어는 줄 바꿈 없이 한 줄에 입력한다. 결과 값에서 id 는 다른 값을 가질 수 있다.

```
# openstack endpoint create --region RegionOne volume public
http://controller:8776/v1/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 8bc6b83649014e7bb303e55b2fc77da2        |
```

```
| interface    | public                                   |
| region       | RegionOne                                |
| region_id    | RegionOne                                |
| service_id   | 7db5c4ba9044469b837cd95d2523e41f         |
| service_name | cinder                                   |
| service_type | volume                                   |
| url          | http://controller:8776/v1/%(tenant_id)s  |
+--------------+------------------------------------------+

# openstack endpoint create --region RegionOne volume internal
http://controller:8776/v1/%\(tenant_id\)s
+--------------+------------------------------------------+
| Field        | Value                                    |
+--------------+------------------------------------------+
| enabled      | True                                     |
| id           | 560daf920c4e4021831d42aeddd4114c         |
| interface    | internal                                 |
| region       | RegionOne                                |
| region_id    | RegionOne                                |
| service_id   | 7db5c4ba9044469b837cd95d2523e41f         |
| service_name | cinder                                   |
| service_type | volume                                   |
| url          | http://controller:8776/v1/%(tenant_id)s  |
+--------------+------------------------------------------+

# openstack endpoint create --region RegionOne volume admin
http://controller:8776/v1/%\(tenant_id\)s
+--------------+------------------------------------------+
| Field        | Value                                    |
+--------------+------------------------------------------+
| enabled      | True                                     |
| id           | c2be1121fc4b4717a50d33dd46f9f15b         |
| interface    | admin                                    |
| region       | RegionOne                                |
| region_id    | RegionOne                                |
| service_id   | 7db5c4ba9044469b837cd95d2523e41f         |
```

```
| service_name | cinder                                    |
| service_type | volume                                    |
| url          | http://controller:8776/v1/%(tenant_id)s |
+--------------+-------------------------------------------+
```

```
# openstack endpoint create --region RegionOne volumev2 public
http://controller:8776/v2/%\(tenant_id\)s
+--------------+-------------------------------------------+
| Field        | Value                                     |
+--------------+-------------------------------------------+
| enabled      | True                                      |
| id           | 7a75a7b47d934d01a6c517a700fece16          |
| interface    | public                                    |
| region       | RegionOne                                 |
| region_id    | RegionOne                                 |
| service_id   | 2e8f964acce84db7bb846df4e235942e          |
| service_name | cinderv2                                  |
| service_type | volumev2                                  |
| url          | http://controller:8776/v2/%(tenant_id)s |
+--------------+-------------------------------------------+

# openstack endpoint create --region RegionOne volumev2 internal
http://controller:8776/v2/%\(tenant_id\)s
+--------------+-------------------------------------------+
| Field        | Value                                     |
+--------------+-------------------------------------------+
| enabled      | True                                      |
| id           | 2952dea8705c4e5c9ab7e5e39b47a8b4          |
| interface    | internal                                  |
| region       | RegionOne                                 |
| region_id    | RegionOne                                 |
| service_id   | 2e8f964acce84db7bb846df4e235942e          |
| service_name | cinderv2                                  |
| service_type | volumev2                                  |
| url          | http://controller:8776/v2/%(tenant_id)s |
+--------------+-------------------------------------------+
```

```
# openstack endpoint create --region RegionOne volumev2 admin
http://controller:8776/v2/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 02f3ba6b30fa48d5a708cce4e829a0fa        |
| interface    | admin                                   |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 2e8f964acce84db7bb846df4e235942e        |
| service_name | cinderv2                                |
| service_type | volumev2                                |
| url          | http://controller:8776/v2/%(tenant_id)s |
+--------------+-----------------------------------------+
```

● 구성요소 설치 및 설정

기본 설정 파일에서 주석 처리되어 있는 항을 그대로 두고 바로 아래에 설정 내용을 추가할 것을 권장한다. 설정 파일에서(...) 부분은 기존 설정파일에서 유지되어야할 디폴트 옵션을 나타낸다. 다음의 모든 과정은 컨트롤러 노드에서 진행한다.

1. 패키지를 설치한다.

```
# yum -y install openstack-cinder python-cinderclient
```

2. /etc/cinder/cinder.conf 파일을 다음과 같이 수정한다.

a. [database] 섹션에서 데이터베이스 접속을 설정한다. 데이터베이스 암호로 *cinderdbpass* 를 사용한다.

```
# vi /etc/cinder/cinder.conf
[database]
...
connection = mysql+pymysql://cinder:cinderdbpass@controller/cinder
```

b. [DEFAULT] 와 [oslo_messaging_rabbit] 섹션에서 RabbitMQ 메시지 큐 접속을 설정한다. rabbit_password 에는 *rabbitpass* 를 사용한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
```

```
...
rpc_backend = rabbit

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = rabbitpass
```

c. [DEFAULT]와 [keystone_authtoken] 섹션에서 자격증명 서비스 접속을 설정한다. password 는 *cinderpass* 를 사용한다. [keystone_authtoken] 섹션의 다른 옵션은 모두 삭제하거나 주석처리한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
auth_strategy = keystone

[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = cinder
password = cinderpass
```

d. [DEFAULT] 섹션에서 my_ip 옵션을 컨트롤러 노드의 관리 인터페이스 주소인 10.0.0.11 로 설정한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
my_ip = 10.0.0.11
```

e. [oslo_concurrency] 섹션에서 lock_path 를 설정한다.

```
# vi /etc/cinder/cinder.conf
[oslo_concurrency]
...
```

```
lock_path = /var/lib/cinder/tmp
```

3. 블록 스토리지 데이터베이스를 채워준다. 경고 메시지는 무시한다.

```
# su -s /bin/sh -c "cinder-manage db sync" cinder
```

● 블록 스토리지 사용을 위한 컴퓨트 설정

/etc/nova/nova.conf 파일을 다음과 같이 수정한다.

```
# vi /etc/nova/nova.conf
[cinder]
os_region_name = RegionOne
```

● 설치 완료

1. 컴퓨트 API 서비스를 재시작한다.

```
# systemctl restart openstack-nova-api.service
```

2. 블록 스토리지 서비스를 시작하고 시스템 시작시에 자동으로 시작되도록 설정한다.

```
# systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service
# systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service
```

4.7.3 스토리지 노드 설치 및 구성

이 섹션에서는 블록 스토리지 서비스를 위해 스토리지 노드를 설치하고 구성하는 과정을 설명한다. 이 가이드에서는 별도의 스토리지 노드를 사용하지 않고 컨트롤러 노드를 스토리지 노드로 사용하므로 VMware Player 구성에서 생성한 cinder 볼륨 저장용 100 GB 의 가상 디스크를 사용한다. 이 가상디스크는 /dev/sdc 로 지정되어 있는데, 파티션 설정이나 포맷이 필요하지 않다. 별도의 스토리지 노드를 사용할 경우에는 비어있는 별도의 로컬 디스크를 준비한다.
이 서비스에서는 LVM 드라이버를 사용하여 디스크에 논리적 볼륨을 구성하고 이를 iSCSI 전송을 통해 인스턴스에 제공한다. 이 절에서 설명하는 내용 중에서 몇 가지를 수정하여 스토리지 노드 추가를 통해 수평확장이 가능하다.

● 사전 준비 사항

스토리지 노드(여기에서는 컨트롤러 노드를 스토리지 노드로 겸한다)에 블록 스토리지 서비스를 설치하기에 앞서 스토리지 디바이스를 준비한다.(이 명령은 스토리지 노드에서 수행한다)

1. 스토리지 서비스 관련 패키지를 설치한다.

a. LVM 패키지를 설치한다.(CentOS 7 에는 기본으로 설치되어 있다)

```
# yum -y install lvm2
```

b. LVM 메타데이터 서비스를 실행하고 시스템 시작시에 자동으로 실행되도록 설정한다.

```
# systemctl enable lvm2-lvmetad.service
# systemctl start lvm2-lvmetad.service
```

2. LVM 물리적 볼륨인 /dev/sdc 를 생성한다.(시스템에 따라 /dev/sdc 는 변경될 수 있다)

```
# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created
```

3. cinder-volumes 라는 이름으로 LVM 볼륨 그룹을 생성한다. 블록 스토리지 서비스는 논리적 볼륨을 이 볼륨 그룹 내에 생성하게 된다.

```
# vgcreate cinder-volumes /dev/sdc
Volume group "cinder-volumes" successfully created
```

4. 블록 스토리지 볼륨에 대해서는 인스턴스만 접근이 가능하다. 그런데 운영체제가 볼륨에 관련된 디바이스를 관리하므로 LVM 볼륨 탐색툴은 기본적으로 볼륨을 포함하고 있는 블록 스토리지 디바이스에 대해서 전체 /dev 디렉토리를 탐색하게 된다. 따라서 프로젝트가 LVM 볼륨을 사용할 경우 탐색툴이 이 볼륨을 감지하고 캐싱하려고 하는데, 이 경우 운영체제와 프로젝트 간에 다양한 문제가 발생할 수 있다. 따라서 LVM 이 cinder-volume 볼륨 그룹에 있는 디바이스만 탐색하도록 /etc/lvm/lvm.conf 파일을 다음과 같이 수정한다.

a. devices 섹션에서 OS 에서 사용하는 /dev/sda 와 블록 스토리지에서 사용하는 /dev/sdc 디바이스만 탐색을 허용하고 나머지 디바이스는 거부하도록 필터를 추가한다. filter 옵션 내의 a 는 허용(accept)을 의미하고 r 은 거부(reject)를 의미하며, 정규식으로 디바이스 이름이 표현된다. r/.*/ 은 나머지 모든 디바이스 탐색을 거부한다는 의미이다.

```
# vi /etc/lvm/lvm.conf
devices {
...
filter = [ "a/sda/", "a/sdc/", "r/.*/"]
```

- 구성 요소 설치 및 구성

1. 패키지를 설치한다.

```
# yum -y install openstack-cinder targetcli python-oslo-policy
```

2. /etc/cinder/cinder.conf 파일을 다음과 같이 수정한다.(컨트롤러 노드를 스토리지 노드로 사용할 경우 a~e 의 과정은 건너뛰고 f 부터 진행한다)

a. [database] 섹션에서 데이터베이스 접속을 설정한다. 데이터베이스 암호로 *cinderdbpass* 를 사용한다.

```
# vi /etc/cinder/cinder.conf
```

```
[database]
...
connection = mysql+pymysql://cinder:cinderdbpass@controller/cinder
```

b. [DEFAULT] 와 [oslo_messaging_rabbit] 섹션에서 RabbitMQ 메시지 큐 접속을 설정한다. rabbit_password 에는 *rabbitpass* 를 사용한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
rpc_backend = rabbit

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = rabbitpass
```

c. [DEFAULT]와 [keystone_authtoken] 섹션에서 자격증명 서비스 접속을 설정한다. password 는 *cinderpass* 를 사용한다. [keystone_authtoken] 섹션의 다른 옵션은 모두 삭제하거나 주석처리한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
auth_strategy = keystone

[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = cinder
password = cinderpass
```

d. [DEFAULT] 섹션에서 my_ip 옵션을 스토리지 노드의 관리 인터페이스 주소인 10.0.0.41 로 설정한다.(컨트롤러 노드를 스토리지 노드로 사용할 경우 10.0.0.11 을 그대로 둔다)

```
# vi /etc/cinder/cinder.conf
```

```
[DEFAULT]
...
my_ip = 10.0.0.41
```

e. [oslo_concurrency] 섹션에서 lock_path 를 설정한다.

```
# vi /etc/cinder/cinder.conf
[oslo_concurrency]
...
lock_path = /var/lib/cinder/tmp
```

f. [lvm] 섹션에 LVM 드라이버, cinder-volumes 볼륨 그룹, iSCSI 프로토콜 및 적절한 iSCSI 서비스를 포함하여 LVM 백엔드를 설정한다.

```
# vi /etc/cinder/cinder.conf
[lvm]
...
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
volume_group = cinder-volumes
iscsi_protocol = iscsi
iscsi_helper = lioadm
```

f. [DEFAULT] 섹션에서 LVM 벡엔드를 활성화한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
enabled_backends = lvm
```

g. [DEFAULT] 섹션에 이미지 서비스 API 가 실행되는 위치를 설정한다.

```
# vi /etc/cinder/cinder.conf
[DEFAULT]
...
glance_api_servers = http://controller:9292
```

- 설치 완료

1. 블록 스토리지 볼륨 서비스를 시작하고 시스템 시작시에 자동으로 시작되도록 설정한다.

```
# systemctl enable openstack-cinder-volume.service target.service
# systemctl start openstack-cinder-volume.service target.service
```

4.7.4 동작 확인

컨트롤러 노드에서 다음의 명령을 수행하여 블록 스토리지 서비스가 정상적으로 동작하는지 확인한다.

1. 관리자 모드로 CLI 명령을 실행하기 위해 admin 인증 스크립트를 실행한다.

```
# source /root/admin-openrc
```

2. 서비스 구성요소를 출력하여 각 프로세스가 정상적으로 실행되었는지 확인한다.

```
# cinder service-list
+------------------+-----------------+------+---------+-------+--------------------
|      Binary      |       Host      | Zone |  Status | State |         Updated_at
+------------------+-----------------+------+---------+-------+--------------------
| cinder-scheduler | controller      | nova | enabled |  up   | 2016-08-14T11:49:07
| cinder-volume    | controller @lvm | nova | enabled |  up   | 2016-08-14T11:49:06
+---- -------------+-----------------+------+---------+-------+--------------------
```