Ceph 스토리지 구축

ceph 소개

1. 분산 객체 스토리지를 구성하는 OSS
2. 서버 구성은 OSD, Monitor, Manager, MDS 서버가 필요
3. 논리적으로 구성한 Storage pool안에서 데이터를 개체로 저장. Crush알고리즘을 사용해서 배치그룹을 계산하고 저장

구성정보

1. Component 종류
   1. ceph-mon(모니터노드) : 클러스터 상태를 체크하고, 데몬과 클라이언트간 인증관리 담당 / HA구성시 3대 필요
   2. ceph-mgr(관리노드) : 스토리지 활용도 / 현재상태 및 메트릭 추적 (dashboard 및 RestAPI 제공) / HA구성시 2대 필요(Active / Standby)
   3. ceph-osd(객체스토리지 데몬) : 데이터를 저장하고 복제 / 부하분산 역활을 수행 (OSD디스크 1TB당 메모리 1G이상으로 구성을 권고), HA구성시 최소 3대 필요
   4. ceph-mds : CEPH FS를 대신해서 메타 데이터를 관리하는 서버. = Block Devices / Object Storage에서는 MDS를 사용하지 않음
2. Component Hardware Spec

Component	Hardward	Spec
osd	CPU	OSD당 2 Core
osd	MEM	데몬당 4GB이상
osd	DISK	최소 1TB이상,(SSD 권장) 단일 디스크에서 여러 OSD 실행은 비권장 단일 디스크에서 osd+mon+mds 실행방식 비권장 OSD용 디스크는 OS와 분리해서 사용(성능저하 이슈)
osd	NIC	10G이상
mon	CPU	2코어 이상
mon	MEM	데몬당 24GB이상
mon	DISK	데몬당 60GB
mds	CPU	2코어 이상
mds	MEM	데몬당 2GB이상
mds	DISK	데몬당 1MB 이상
mds	NIC	1Gb 이상

* OSD에 RAID구성시 성능저하가 발생할 수 있으므로 BMT를 통해 성능 비교 권고

시스템 이해

1. OSD Backend
   1. Bluestore
      - Ceph 12.2이후 부터 default storage
      - 저장장치를 직접 엑세스 해서 데이터를 관리 - XFS같은 파일시스템을 사용하지 않음
      - RocksDB를 통한 메타데이터 관리
      - 전체 데이터 및 메타데이터 checksum 수행 - 무결성 유지
      - inline압축 - 디스크에 저장하기 전에 선택적으로 압축수행
      - 데이터 관리 계층화 - journal을 별도 장치에 기록할 수 있어 성능향상 가능.
      - CoW을 사용하기 때문에 기존보다 향상된 IO
   2. Filestore

- Ceph에 개체를 저장하는 방식.
- 일부 메타데이터에 대해 LevelDB를 사용해 key/value로 저장
- 파일시스템을 btrfs / ext4에서 사용시 알려진 결함이 있어 데이터가 손실될 수 있음 (XFS는 영향없음)

2. Pool
   1. 개체를 저장하기 위해 사용하는 논리 파티션
      - Recovery : 데이터 손실없이 사용할 수 있도록 설정하는 OSD
      - PG : Pool에 대한 배치 그룹 수 (일반적으로 OSD당 100개의의 PG을 사용)
      - Cursh Rule : 데이터를 Pool에 저장할때 Crush Rule에 의해 결정
      - Snapshot : 특정 Pool의 스냅샷 생성
   2. Pool을 사용하기 위해서는 어플리케이션과 연결되어 있어야 하며, RBD에서 사용할 경우 RBD도구를 사용해서 초기화가 필요 (cephfs / rbd / rgw 중 택1)
3. CephFS
   1. 분산 개체 저장소인 RADOS를 기반으로 구축된 파일시스템
   2. 공유 디렉토리 및 HA를 제공
   3. CephFS는 데이터용와 메다데이터용으로 각각 2개이상의 RAODS Pool이 필요
      - 메타데이터 pool에서 데이터가 손실되면 전체파일 시스템 액세스가 불가능
      - 메타 데이터 pool에 SSD 사용
      - 데이터 풀은 파일시스템을 생성하고, 기본적으로 모든 inode 정보를 저장하는 위치
4. NFSExport
   1. NFS-Ganesha NFS를 이용해 CephFS 네임스페이스 export 가능

Ceph 설치하기 (ansible 기반의 ceph배포)

1. 설치 방법에는 cephadm / Rook / ansible을 이용한 설치방법이 존재,
   1. cephadm - 자체적으로 설치하는 binary container 혹은 python3이 필요
   2. Rook - kuernetes에서 ceph를 설치하거나 기존 ceph를 k8s로 join할때 Rook을 이용
   3. ceph-deploy은 최신버전에서 사용되지 않음
2. ceph-ansible을 설치하기 위한 python 패키지 설치

```
$ yum install -y python3 python3-pip sshpass
$ pip3 install --upgrade setuptools pip --ignore-installed
```

3. ceph-ansible 내려받기

```
$ git clone https://github.com/ceph/ceph-ansible.git -b "v6.0.13"
$ cd ceph-ansible
```

- ceph-ansible 버전별 대응 버전

ceph-ansible	ceph	ansible
3.0	jewel / luminous	2.4
3.1	luminous / mimic	2.4
3.2	luminous / mimic	2.6
4.0	nautilus	2.9
5.0	octopus	2.9
6.0	pacific	2.9

4. dependency 패키지 설치

```
$ pip3 install -r requirements.txt
```

5. 배포를 위한 호스트파일 작성

```
$ vi hosts

[mons]
192.168.100.41

[osds]
```

```
192.168.100.41
192.168.100.42

[mdss]

[rgws]

[nfss]
192.168.100.41

[rbdmirrors]

[clients]
192.168.100.41

[mgrs]
192.168.100.41

[iscsigws]

[iscsi-gws]

[grafana-server]

[rgwloadbalancers]

[monitoring]
192.168.100.41

[all:vars]
ansible_become=true
ansible_user=root
ansible_ssh_pass=root
```

6. 환경변수 복사 (systemd 기반으로 구동시)

```
$ cp site.yml.sample site.yml
$ cp group_vars/all.yml.sample  group_vars/all.yml
$ cp group_vars/osds.yml.sample  group_vars/osds.yml
```

7. 환경변수 복사 (container 기반으로 구동시)

```
$ cp site-container.yml.sample  site.yml
$ cp group_vars/all.yml.sample  group_vars/all.yml
$ cp group_vars/osds.yml.sample  group_vars/osds.yml
```

8. config 설정 (systemd 기반으로 구동시)

```
$ vi group_vars/all.yml
...
osd_objectstore: bluestore
monitor_interface: ens3f0
public_network: 192.168.100.0/24
ntp_service_enabled: true
ntp_daemon_type: chronyd
...
##############
# DASHBOARD #
##############
dashboard_enabled: false
dashboard_protocol: http
dashboard_port: 8081
dashboard_admin_user: admin
dashboard_admin_password: adminpassword
containerized_deployment: false
...
configure_firewall: false
...
ceph_origin: repository
...
```

```
ceph_repository: community
...
ceph_stable_release: octopus
```

```
$ vi group_vars/osds.yml
...
devices:
  - /dev/sdb
...
```

```
$ vi roles/ceph-validate/tasks/main.yml
...
  #해당 name 전체 삭제
  - name: validate ceph_repository_community
    fail:
      msg: "ceph_stable_release must be 'pacific'"
    when:
      - ceph_origin == 'repository'
      - ceph_repository == 'community'
      - ceph_stable_release not in ['pacific']
...
```

Centos7에서 systemd 기반으로 구동시 dashboard가 호환되지 않아 false로 처리해야 함
ceph 릴리즈 버전 중 pacific 버전은 Centos7에서 nfs export가 되지 않아 octopus로 다운그레이드가 필요
config 설정 (ceph를 container로 구동시)

```
$ vi group_vars/all.yml
...
osd_objectstore: bluestore
monitor_interface: ens3f0
public_network: 192.168.100.0/24
ntp_service_enabled: true
ntp_daemon_type: chronyd
...
##############
# DASHBOARD #
##############
dashboard_enabled: false
containerized_deployment: true
...
```

```
$ vi group_vars/osds.yml
...
devices:
  - /dev/sdb
...
```

9. 배포

```
$ ansible-playbook  -i hosts  site.yml -b -v
```

10. cluster health check시 warn 발생시
    #Cluster 구성상태 모두 정상인데, health check warn으로 표시될 경우 조치방법 (ceph자체 버그로 의심)

```
$ ceph config set mon auth_allow_insecure_global_id_reclaim false
```

운영방법

1. ceph cluster상태 확인

```
$  ceph status
  cluster:
    id:     ca96d48d-1c9d-4168-9f21-ffda54a5cd9c
    health: HEALTH_OK

  services:
    mon: 2 daemons, quorum openstack-dev1,openstack-dev2 (age 87m)
    mgr: openstack-dev1(active, since 78m), standbys: openstack-dev2
    osd: 3 osds: 3 up (since 83m), 3 in (since 2h)

  data:
    pools:   5 pools, 105 pgs
    objects: 49 objects, 5.3 KiB
    usage:   41 MiB used, 300 GiB / 300 GiB avail
    pgs:     105 active+clean
```

2. ceph osd 상태 확인

```
$ ceph osd tree
ID  CLASS  WEIGHT   TYPE NAME             STATUS  REWEIGHT  PRI-AFF
-1         0.29306  root default
-5         0.09769      host dev1
2    hdd  0.09769          osd.2             up   1.00000  1.00000
-3         0.09769      host dev2
0    hdd  0.09769          osd.0             up   1.00000  1.00000
-7         0.09769      host dev3
1    hdd  0.09769          osd.1             up   1.00000  1.00000
```

3. ceph현재 latency 확인방법

```
$ ceph osd perf
osd  commit_latency(ms)  apply_latency(ms)
 2                   0                  0
 0                   0                  0
 1                   0                  0
```

commit은 시스템 call이 있기 때문에 일반적으로 100 ~ 600ms까지는 수용가능한 수준으로 판단
메모리내 적용된 파일을 파일시스템에 적용하는 시간 (ms단위, 실제 성능에 판단되는 시간)

4. nfs 오류시 로그 확인

```
$ cephadm logs --fsid <fsid> --name nfs.{{ clusteid }}.hostname
```

5. 파일시스템1. CephFS - Pool 관리

```
$ ceph osd lspools
```

1. Pool 생성
   #Pool 생성

```
$ ceph osd pool create {{ DATA_POOL_NAME }}
$ ceph osd pool create {{ METADATA_POOL_NAME }}
```

#CephFS는 데이터용과 메타데이터용 각각 2개이상의 RADOS풀 필요

2. 생성된 Pool을 애플리케이션에 연결 (cephfs로 연결)

```
$ ceph osd pool application enable {{ DATA_POOL_NAME }} cephfs
```

3. 파일시스템 생성

```
$ ceph fs new {{ FS_NAME }} {{ METADATA_POOL_NAME  }} {{DATANAME }}
```

4. NFS export

   # 1. nfs module설정

   ```
   $ ceph mgr module enable nfs
   ```

   # 2. nfs ganesha 클러스터 생성

   ```
   $ ceph nfs cluster create {{ clusterid }}
   ```

   # 3. nfs export

   ```
   $ ceph nfs export create cephfs {{ NAME }} {{ clusterid }
   ```

reference

- https://docs.ceph.com/en/latest/architecture/
- *https://www.slideshare.net/jenshadlich/ceph-object-storage-at-spreadshirt-july-2015-ceph-berlin-meetup*

Revision #2
★Created 8 June 2022 03:37:43 by artop0420
✎Updated 11 January 2024 04:37:31 by artop0420