목차

구축사례 - Ceph RBD with Samba HA (Pacemaker,LVM)

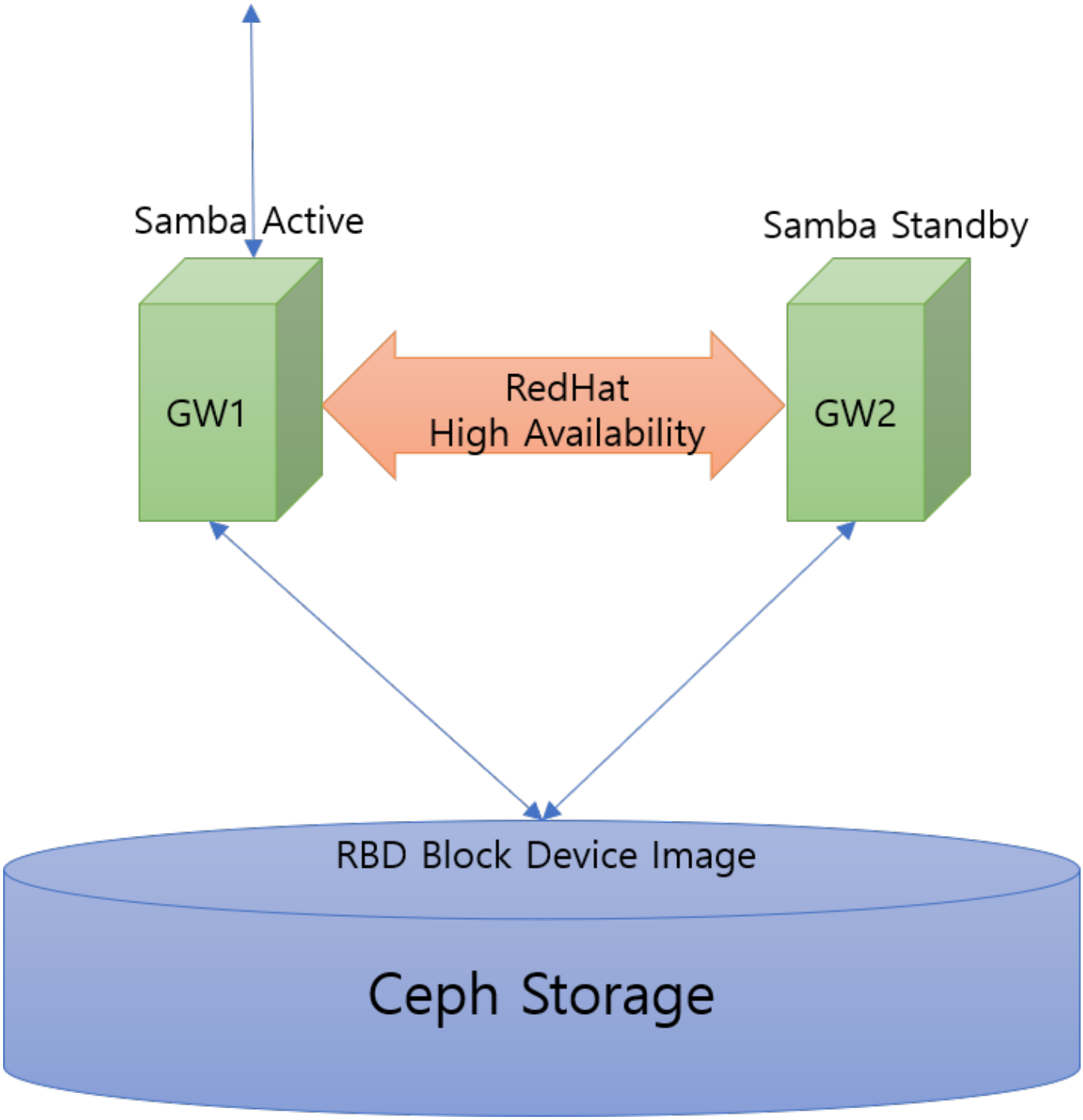
— *이강우 2018/12/06 13:25*

본 문서는 Ceph RBD를 BackStore Storage 로 사용하고 Samba를 통해 CIFS 서비스를 HA로 구성하는 사례를 설명한다.

구성

- RedHat Enterprise Linux 7.6 with High Availability
- RedHat Ceph Storage 3.1 (Ceph Community 12.2.5 Luminous)

전체적인 구성도는 아래와 같다.



Ceph Storage에서 RBD로 블럭장치를 제공받아 해당 블럭에 LVM파일시스템을 생성하여 그것을 Samba CIFS로 제공하는 구성이다. 이 구성은 간단한 구성이며 NFS같은 다른 서비스에도 동일한 형태로 구성이 가능하다.

단지 차이점이 있다면 고가용성(High Availability)제공 방법을 어떤것으로 구성하느냐 하는점에 차이가 있을수 있다. 본 사례에서는 RedHat High Availability(pacemaker)를 사용하여 구성하지만 상황에 따라 CTDB, Ganesha, HAProxy등 다른 솔루션을 사용하여 구축하는 방법도 가능할것으로 본다.

Ceph RBD 구성

일단 Ceph Storage 자체는 이미 구성되어있다고 가정한다. 이 부분은 별도로 Ceph 관련 문서를 참조하

기 바란다. 구성되어있는 Ceph Storage에서 Samba CIFS 서비스 제공을 위한 공유 스토리지 용도의 블럭 장치 이미지를 생성한다.

```
[root@ceph-mon1 ~]# ceph osd pool create rbd 64 64
pool 'rbd' created

[root@ceph-mon1 ~]# ceph osd pool application enable rbd rbd
enabled application 'rbd' on pool 'rbd'

[root@ceph-mon1 ~]# ceph osd pool ls detail
pool 1 '.rgw.root' replicated size 3 min_size 2 crush_rule 0 object_hash
rjenkins pg_num 8 pgp_num 8 last_change 28 flags hashpspool stripe_width 0
application rgw
pool 2 'default.rgw.control' replicated size 3 min_size 2 crush_rule 0
object_hash rjenkins pg_num 8 pgp_num 8 last_change 30 flags hashpspool
stripe_width 0 application rgw
pool 3 'default.rgw.meta' replicated size 3 min_size 2 crush_rule 0
object_hash rjenkins pg_num 8 pgp_num 8 last_change 32 flags hashpspool
stripe_width 0 application rgw
pool 4 'default.rgw.log' replicated size 3 min_size 2 crush_rule 0
object_hash rjenkins pg_num 8 pgp_num 8 last_change 34 flags hashpspool
stripe_width 0 application rgw
pool 5 'rbd' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 64 pgp_num 64 last_change 84 flags hashpspool stripe_width 0
application rbd
```

위와같이 rbd pool 이 생성되었다. 해당 rbd pool에 10GB짜리 공유 스토리지용 볼륨이미지를 생성한다.

```
[root@ceph-mon1 ~]# rbd create cifsvol --size 10G
[root@ceph-mon1 ~]# rbd ls -l
NAME      SIZE PARENT FMT PROT LOCK
cifsvol 10240M          2
```

여기까지 Ceph storage 에서의 준비는 끝났다. 스토리지쪽에서 해줘야 할일은 단순히 공유볼륨을 만들어서 제공하는것뿐이다.

Gateway 서버 구성

Samba 서비스를 제공할 Gateway서버를 구성한다. 고가용성을 위해 서버 2대를 RedHat High Availability 로 구성하도록 한다.

Gateway 서버는 아래 2대로 구성되었다고 가정한다.

- rhcs1.local.domain
- rhcs2.local.domain

Pacemaker로 두 서버를 클러스터로 구성해준다.

```
[root@rhcs1 cib]# pcs cluster auth rhcs1.local.domain rhcs2.local.domain
rhcs2.local.domain: Already authorized
rhcs1.local.domain: Already authorized

[root@rhcs1 cib]# pcs cluster setup --start --name HACL rhcs1.local.domain
rhcs2.local.domain
Destroying cluster on nodes: rhcs1.local.domain, rhcs2.local.domain...
rhcs1.local.domain: Stopping Cluster (pacemaker)...
rhcs2.local.domain: Stopping Cluster (pacemaker)...
rhcs1.local.domain: Successfully destroyed cluster
rhcs2.local.domain: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'rhcs1.local.domain',
'rhcs2.local.domain'
rhcs1.local.domain: successful distribution of the file 'pacemaker_remote
authkey'
rhcs2.local.domain: successful distribution of the file 'pacemaker_remote
authkey'
Sending cluster config files to the nodes...
rhcs1.local.domain: Succeeded
rhcs2.local.domain: Succeeded

Starting cluster on nodes: rhcs1.local.domain, rhcs2.local.domain...
rhcs1.local.domain: Starting Cluster (corosync)...
rhcs2.local.domain: Starting Cluster (corosync)...
rhcs1.local.domain: Starting Cluster (pacemaker)...
rhcs2.local.domain: Starting Cluster (pacemaker)...

Synchronizing pcsd certificates on nodes rhcs1.local.domain,
rhcs2.local.domain...
rhcs2.local.domain: Success
rhcs1.local.domain: Success
Restarting pcsd on the nodes in order to reload the certificates...
rhcs2.local.domain: Success
rhcs1.local.domain: Success

[root@rhcs1 cib]# pcs cluster enable --all
rhcs1.local.domain: Cluster Enabled
rhcs2.local.domain: Cluster Enabled

[root@rhcs1 cib]# pcs status
Cluster name: HACL

WARNINGS:
No stonith devices and stonith-enabled is not false

Stack: corosync
Current DC: rhcs2.local.domain (version 1.1.19-8.el7_6.1-c3c624ea3d) -
partition with quorum
Last updated: Tue Dec  4 17:51:27 2018
```

```
Last change: Tue Dec  4 17:39:03 2018 by hacluster via crmd on
rhcs2.local.domain

2 nodes configured
0 resources configured

Online: [ rhcs1.local.domain rhcs2.local.domain ]

No resources


Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

이렇게 하면 일단 두 노드간 클러스터 구성은 완료되었다.

RBD 블럭장치 공유볼륨 구성

이제 Ceph Storage에서 제공한 볼륨을 공유볼륨으로 구성하도록 한다.

Ceph Storage에서 생성한 공유볼륨 이미지인 `cifsvol` 이미지를 양쪽노드에서 사용할수 있도록 설정한다. 본 사례에서는 RBD Image를 사용하기 위해 `rbdmap` 서비스를 이용한다.

/etc/ceph/rbdmap

```
# RbdDevice     Parameters
rbd/cifsvol    id=admin,keyring=/etc/ceph/ceph.client.admin.keyring
```

실제로 장치가 잘 연결되는지 확인해본다.

```
# systemctl start rbdmap
# lsblk
[root@rhcs1 ceph]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   20G  0 disk
├─sda1                8:1    0    1G  0 part /boot
└─sda2                8:2    0   19G  0 part
  ├─rhel_rhcs1-root 253:0    0   17G  0 lvm  /
  └─rhel_rhcs1-swap 253:1    0    2G  0 lvm  [SWAP]
sdb                   8:16   0   20G  0 disk
└─sdb1                8:17   0   20G  0 part
sr0                  11:0    1  4.2G  0 rom
```

```
rbd0                  252:0    0   10G  0 disk
```

위와같이 정상적으로 rbd0 장치가 보이게 된다. 이제 이 rbd0 장치에 LVM파티션을 생성한다. LVM관련 작업내용은 생략한다.

```
[root@rhcs1 ceph]# lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                     8:0    0   20G  0 disk
├─sda1                  8:1    0    1G  0 part /boot
└─sda2                  8:2    0   19G  0 part
  ├─rhel_rhcs1-root   253:0    0   17G  0 lvm  /
  └─rhel_rhcs1-swap   253:1    0    2G  0 lvm  [SWAP]
sdb                     8:16   0   20G  0 disk
└─sdb1                  8:17   0   20G  0 part
sr0                    11:0    1  4.2G  0 rom
rbd0                  252:0    0   10G  0 disk
└─rbd0p1              252:1    0   10G  0 part
  └─vg_rbd-lv_rbd     253:2    0   10G  0 lvm  /rbd    <-- 생성되었다
```

여기까지 공유볼륨 준비 과정은 끝났다. 공유볼륨 설정을 하기 위해 일단 볼륨을 umount 하도록 하자.

```
# umount /rbd
# vgchange -an vg_rbd
# systemctl stop rbdmap

# lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                     8:0    0   20G  0 disk
├─sda1                  8:1    0    1G  0 part /boot
└─sda2                  8:2    0   19G  0 part
  ├─rhel_rhcs1-root   253:0    0   17G  0 lvm  /
  └─rhel_rhcs1-swap   253:1    0    2G  0 lvm  [SWAP]
sdb                     8:16   0   20G  0 disk
└─sdb1                  8:17   0   20G  0 part
sr0                    11:0    1  4.2G  0 rom
```

그리고 LVM설정을 공유볼륨을 사용할수 있도록 변경한다. 본 사례에서는 공유볼륨을 HA LVM(Tagging)을 이용하는 방법을 사용하므로 아래와 같이 설정해준다. HA LVM방식이기 때문에 Active/Standby 형태로만 서비스 가능하다.

```
# lvmconf --enable-halvm --services --startstopservices
```

이후 volume_list 를 변경하고 dracut 을 수행하여 initramfs 장치를 업데이트 해준다.

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availabilit

y_add-on_administration/s1-exclusiveactive-haaa

만약 Active/Active 구성으로 하고자 한다면 clvmd를 사용해야 하며 이 경우 GFS2로 구성하여야 한다.

이제 pacemaker를 통해 리소스 설정을 하도록 한다.

Pacemaker resource 구성

앞부분에서 Pacemaker를 통해 2대의 게이트웨이 서버간 클러스터를 구성했다. 이부분에서는 이렇게 구성된 클러스터에 공유볼륨 리소스를 등록하여 서비스 하도록 한다.

```
# pcs resource create rbdmap systemd:rbdmap --group RBDGRP
# pcs resource create rbdvg LVM volgrpname=vg_rbd exclusive=true --group
RBDGRP
# pcs resource create rbdfs Filesystem  device="/dev/vg_rbd/lv_rbd"
directory="/rbd" fstype="ext4" --group RBDGRP

# pcs resource
 Resource Group: RBDGRP
     rbdmap (systemd:rbdmap):    Started rhcs1.local.domain
     rbdvg  (ocf::heartbeat:LVM):    Started rhcs1.local.domain
     rbdfs  (ocf::heartbeat:Filesystem):    Started rhcs1.local.domain
```

참조문서

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_administration/index
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/logical_volume_manager_administration/index

본 사례에 대한 문의는 문서 상단의 이메일 연락처로 문의해주시기 바랍니다.