Technical Guide

클라우드 환경에서 가상 서버 이중화

SBD Fencing with Pacemaker

May 2021

SAMSUNG SDS

Contents

1. 개요

본 문서는 클라우드 환경에서 가상 서버의 고 가용성을 확보하기 위해 이중화 솔루션(Pacemaker)을 활용하여 서버를 이중화 구성할 때 활용할 수 있는 내용을 다루고 있습니다.

2 개 노드 클러스터 구성에서 Split-brain 현상[1]을 방지하기 위해 Fencing[2] 설정을 하게 되는데 클라우드 환경에서 구성할 수 있는 SBD 를 활용한 Fencing 설정 중에서 주요 옵션과 구성 이후 테스트 항목에 대해 설명합니다.

2. SBD Fencing

SBD Fencing 은 Storage-Based Death 또는 STONITH[3] Block Device 라고 부릅니다. 클러스터 노드 간에 설치된 공유 스토리지를 사용하고, 서버의 sbd 데몬 프로세스가 이 공유 영역에 기록된 메시지를 통해서 Fencing 여부를 결정합니다. Split-brain 상황에서 Fencing 될 노드에 메시지를 보내면 watchdog 에 의해 해당 노드가 스스로 Fencing 하게 됩니다.

SBD 사용 정책(옵션 설정)에 따라 SBD 용 공유 디스크 Fail 시 단독으로 Self-Fencing 하게 설정할 수 있고, SBD 디스크가 Fail 이어도 Pacemaker 의 노드 상태에 따라 Fencing 여부를 결정하도록 할 수도 있습니다.

SBD Fencing 은 Hardware Watchdog 이 더 안정적입니다. 클라우드 환경에서 SBD Fencing 을 위해 Software Watchdog 으로 사용할 수 있지만, 커널 Hang 등의 문제로 인해 정상적으로 처리되지 않을 수도 있습니다.

3. SBD 구성 요소

SBD Fencing 설정을 위해 필요한 구성 요소는 다음과 같습니다.

- SBD 파티션: 노드 간 통신할 수 있는 공유 데이터 영역, 최대 3개까지 추가 가능
- SBD 데몬 프로세스: SBD 영역에 대해 모니터링하고, 클러스터 데몬과 통신
- 메시지: 노드 간 SBD 영역에 메시지를 통해 통신
- Watchdog: Self-Fencing을 위해 주기적으로 감시

[1] 클러스터로 구성된 두 시스템 그룹 간 네트워크의 일시적 단절이 발생 시 나타나는 현상이며, 클러스터 상의 모든 노드들이 노드 각자가 자신을 Primary라고 인식하게 되는 상황

[2] 시스템 장애로부터 데이터를 보호하기 위한 장치/방법으로, OS 리소스 및 HA 클러스터 장애 시 해당 노드의 공유 자원(예: 공유 스토리지)에 대한 연결을 끊어 공유 데이터의 무결성을 보장하는 방법

[3] Shoot The Other Node In The Head

4. SBD Fencing 동작 방식

SBD Fencing 이 어떻게 동작하는지를 살펴보겠습니다.

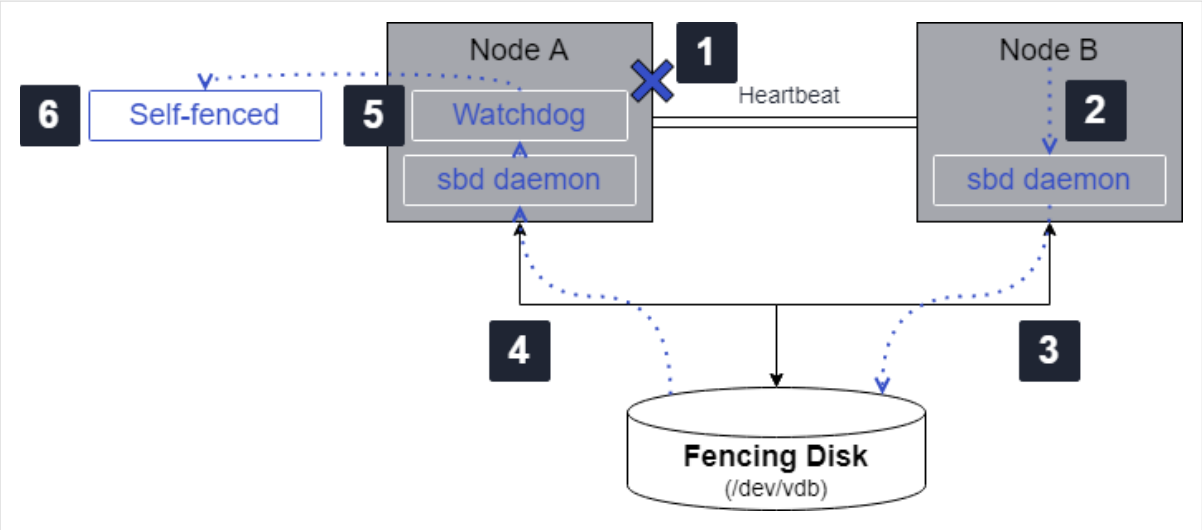


Figure 1. SBD Fencing 동작 방식

1. Node A의 NIC down으로 Heartbeat 통신이 중단되면서 Split-brain 현상 발생
2. Node B의 Pacemaker 데몬 프로세스가 sbd 데몬 프로세스에게 Heartbeat 통신 중단으로 fencing 명령 수행
3. Node B의 sbd 데몬 프로세스는 SBD 디스크에 fencing 메시지 기록
4. Node A의 sbd 데몬 프로세스는 fencing 메시지를 읽고
5. sbd 데몬 프로세스가 watchdog 호출
6. watchdog에 의해 Self-Fencing으로 서버 재부팅

5. SBD Fencing 구성 옵션

SBD Fencing 구성을 위한 주요 옵션에 대한 설명과 설정 값에 대해 알아보겠습니다.

1. Pacemaker는 /etc/sysconfig/sbd 파일에 옵션을 설정합니다.
2. 주요 옵션에 대한 설명은 다음의 표의 내용을 참고합니다.

옵션	설명	권고값
SBD_PACEMAKER (-P)	✓ SBD 디스크가 Fail 되더라도 Pacemaker 의 노드 상태가 정상이면 Fencing 이 발생하지 않음 ✓ IO Fencing 목적이 아닌 일반적인 클러스터 환경에서는 기본적으로 yes 사용을 권고	yes
SBD_WATCHDOG (-W)	✓ watchdog 을 이용해 Fencing	yes
SBD_WATCHDOG_TIMEOUT	✓ - SBD watchdog 장치가 연결이 되지 않을 경우에 대한 타임아웃 시간으로, 이 시간이 지나면 노드를 Fencing 시킴 단, SBD_PACEMAKER=yes 로 설정되고 노드 상태가 정상이면 Fencing 이 발생하지 않음 ✓ watchdog timeout 시간은 스토리지 IO delay 시간에 따라 결정함. ✓ SBD 장치가 multipath 또는 iSCSI 일 경우 시간 초과는 경로 오류를 감지하여 다음 경로로 전환하는 데 필요한 시간으로 설정함 multipath 의 경우, max_polling_interval (default: 4 * polling_interval - 10~20) 보다 크게 설정함	10 ~ 20
msgwait timeout (-4 <timeout>)	✓ SBD 장치의 노드 슬롯에 기록된 메시지가 전달된 것으로 간주되는 시간 ✓ SBD_WATCHDOG_TIMEOUT 의 2 배로 설정함 ✓ Pacemaker 의 stonith-timeout(전체 STONITH 작업이 완료될 때까지 대기하는 시간)은 (msgwait timeout + 20%)값 보다 크게 설정해야 함 stonith-timeout >= msgwait timeout + msgwait timeout * 20%	20 ~ 40

Table 1. SBD Fencing 주요 옵션

3. /etc/sysconfig/sbd 파일 내에 환경변수로 정의가 되지 않았을 경우, SBD_OPTS에 추가로 옵션을 설정할 수 있습니다. (위의 주요 옵션이 파일 내에 있는지 확인하고 없으면 추가합니다.)

```
# cat /etc/sysconfig/sbd

SBD_DEVICE="/dev/vdi"

# Whether to enable the pacemaker integration.
#
SBD_PACEMAKER=yes
# -P 옵션 (해당 옵션이 한 번일 경우 Enable, 두 번(-P -P) 또는 없을 경우 Disable)
SBD_STARTMODE=always
# -S 0
SBD_DELAY_START=no
SBD_WATCHDOG_DEV=/dev/watchdog
SBD_WATCHDOG=yes
# -W 옵션 (해당 옵션이 한 번일 경우 Enable, 두 번(-W -W) 또는 없을 경우 Disable)
SBD_WATCHDOG_TIMEOUT=10

## Type: string
## Default: ""
#
# Additional options for starting sbd
#
SBD_OPTS=""
# 추가 옵션
```

Figure 2. /etc/sysconfig/sbd 파일

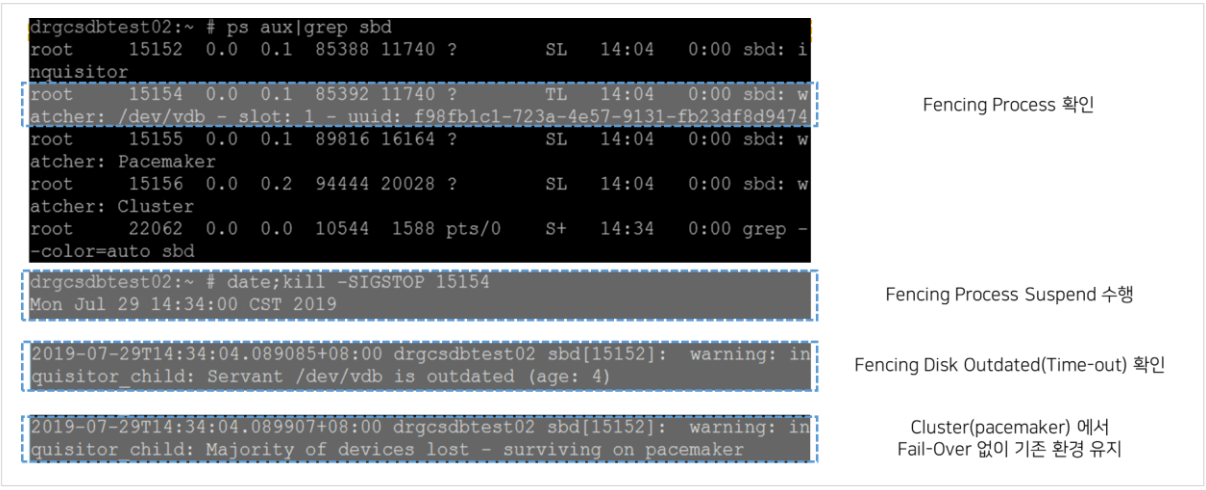
6. SBD Fencing 테스트 결과

SBD Fencing 을 활용하여 Failover 테스트를 수행하고 제대로 동작하는지를 확인합니다.

1. Test Case 1. Primary VM 강제 재부팅 시 Failover 검증

2. Test Case 2. Fencing 디스크 제거를 통한 Failover 검증

3. Test Case 3. Fencing Process Suspend 상황을 통한 Failover 검증



이상으로 SBD 를 활용한 Fencing 구성에 대한 주요 옵션과 테스트 결과를 살펴봤습니다. 클라우드 환경에서 가상 서버에 대한 이중화 구성 시 SBD Fencing 을 활용해 Split-brain 현상을 방지하는데 도움이 되길 바랍니다.