

19장. AHM

19.1. 제품 소개

AHM은 PostgreSQL, AgensSQL과 AgensGraph 데이터베이스 서버에서 발생하는 SPOF (Single point of failure)를 해결하기 위해 Bitnine에서 개발한고가용성 컴포넌트입니다. AHM은 분산 메커니즘을 사용하여 Primary 또는 Standby 데이터베이스의 장애를 감지하고 서비스 연속성을 보장하기 위해 자동으로 적절한 조치를 취합니다.

AHM은 클라이언트 응용 프로그램에 대한 데이터베이스 서버의 가용성을 보장하는 프로그램으로, 각 Primary 및 Standby 데이터베이스와 함께 설치됩니다.

Primary 데이터베이스 및 시스템에 장애가 발생할 경우 AHM은 서로 조정하여 사용 가능한 Standby 데이터베이스 서버 중 하나를 새 Primary 데이터베이스로 지정 한 뒤 이전 Primary 서버에서 VIP(가상 IP)를 기동하여 빠르게 서비스를 정상화 합니다.

19.1.1 Features

- Failover 관리
- 클라이언트 응용 프로그램에 단일 접속 제공(가상 IP 사용)
- 각 Failover 관련 결정에 대해 합의를 구축
- Database and AHM Health check
- 클러스터 상태 관리
- User가 제공한 Pre 및 Post 스크립트 실행

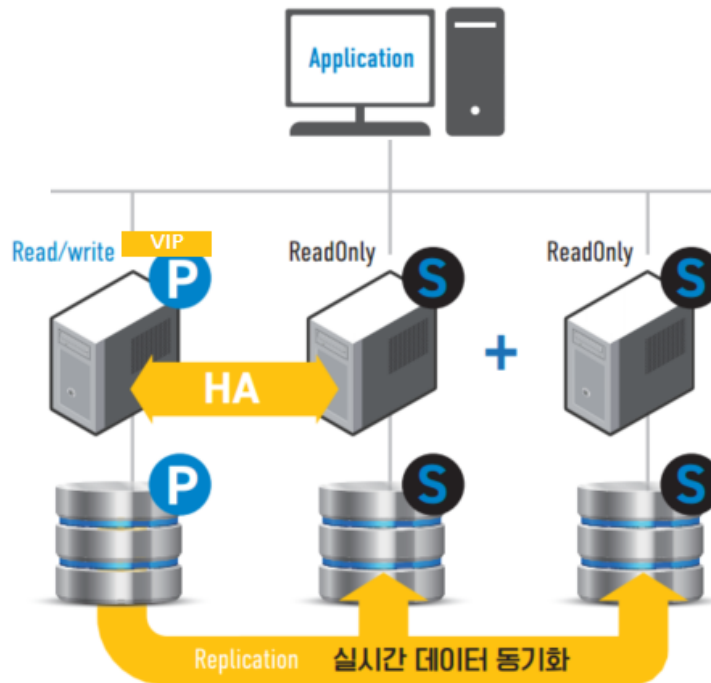
19.1.2 AHM Cluster 구성 요소

- Primary AgensSQL 서버
- 1 또는 그 이상의 스트리밍 복제된 Standby AgensSQL 서버
- 각 Primary 및 Standby 서버에 AHM
- PRIMARY IP, STANDBY IP, VIP 3개의 IP 필요
- 0 또는 그 이상의 AHM witness 노드

AgensSQL Standby 서버 수는 가용성 및 재해 복구 요구 사항에 따라 달라집니다. AHM은 쿼럼의 다수결 메커니즘을 사용하여 리더를 선택하고 Failover를 수행하기 때문에 가용성 요구 사항 및 필수 데이터 보존 정책에 따라 홀수의 AHM 노드를 사용하는 것이 좋습니다.

-일반적인 **AHM** 클러스터

일반적인 AHM 클러스터는 AgensSQL Primary 와 Standby 서버로 구성되어 있습니다.



19.2 AHM Architecture

AHM 노드가 기동 되면 로컬 데이터베이스에 연결하여 데이터베이스 상태를 확인합니다.

- AHM이 데이터베이스에 연결할 수 없는 경우, 리더가 선정될 수 없습니다.
- 데이터베이스가 복구모드로 확인되면 AHM은 Standby 역할을 수행합니다.
- 데이터베이스가 복구모드가 아닌 경우 AHM은 Primary 역할을 수행합니다.
- Primary 데이터베이스가 있는 AHM 노드는 AHM 클러스터의 과반수가 동의하면 VIP를 획득합니다.
- Primary AHM은 Standby 노드의 우선 순위에 따라 Successor 노드를 선택합니다.
- 리더에 장애가 발생 할 경우 Successor 노드는 새로운 리더가 됩니다.
- Failover가 발생할 경우 새로 선택된 AHM 노드는 로컬 데이터베이스 서버에서 노드 Promotion을 수행합니다.

**** Promotion :** Standby로 대기상태에 있는 Database서버가 Primary 로 활성화 되는 과정

19.3 AHM 설치

AHM은 Linux에서 실행됩니다. AHM을 설치 파일은 Bitnine 홈페이지(www.bitnine.net)에 접속하여 설치 파일을 다운로드하여 진행 할 수 있습니다.

19.3.1 AHM 설치 최소 사양

CPU	1 GHz 이상 프로세서
RAM	2GB 이상
DISK	1GB 이상 여유 공간 (AgensSQL설치 파일을 제외한 DATA 공간이 필요합니다)
OS	CentOS 7 이상

19.3.2 사전 빌드된 바이너리 파일을 사용하여 설치

1. AHM 설치 User 생성하기
useradd agens
2. user 암호 설정
passwd agens
3. 생성된 agens 계정으로 Binary 압축 해제
\$ tar -xzvf AHM_1.0.tar.gz
4. 환경 파일 설정
export PATH=/home/agens/AHM_1.0/bin:\$PATH
5. 환경 파일 적용
\$ source ~/.bash_profile

19.4 AHM 구성

AHM는 yaml 형식의 AHM 구성 파일이 필요합니다. 기본적으로 `/usr/local/etc/` directory에서 `ahm_config.yml` 파일을 불러옵니다. 또는 `ahm` 바이너리에 `-f` 명령어를 사용하여 구성 파일 경로를 제공할 수 있습니다.

19.4.1 ahm_config 파라미터 정보

파라미터 명	설명
conn_string	로컬 PG 서버에 대한 연결정보
node_name	AHM 노드 이름, postgresql.conf conninfo의 application_name과 동일하게 설정
nodes_filename	AHM 노드 목록을 파일(AHM_HOME/etc/nodes.conf) 경로
is_witness	Witness 노드여부
priority	Fail Over시 사용되는 노드 우선 순위 (낮을수록 우선)
ahm_port	AHM 노드에 대한 TCP/IP 수신 포트 (기본 포트 : 9500)
ipc_port	IPC에 대한 수신 포트 (기본 포트 : 9595)
data_dir	AHM 데이터 디렉토리 (/tmp)
socket_dir	AHM 소켓파일이 생성될 경로 (/tmp)
log_filename	로그 파일 이름
log_prefix	로그 표기 방식
log_verbose	상세 로그 출력을 활성화합니다
health_check_period	Health check 주기 (단위 : 초)
max_consec_failures	최대 연속 health check 실패 대기 횟수(이후 FailOver 진행)
db_user	데이터베이스 접속 사용자명
db_database	데이터베이스명
db_port	AgensSQL 포트

db_bin_dir	ag_ctl이 포함된 디렉토리 (\$AGHOME/bin)
db_data_dir	데이터 디렉토리 (\$PGDATA)
db_config_dir	데이터베이스 configuration 디렉터리
pre_Promotion_command	FailOver 전에 실행할 스크립트
Promotion_command	FailOver 사용자 정의 명령어
post_Promotion_command	FailOver 후 실행할 스크립트
virtual_IP	VIP 주소
if_cmd_path	If 파일 경로 (/bin)
if_up_cmd	VIP 기동 명령어
if_down_cmd	VIP 정지 명령어
arping_cmd	arping 명령어
arping_path	arping 파일 경로 (/usr/sbin)
ping_path	ping 파일 경로
follow_command	Primary 변경시 Follow Primary 명령어
service_restart_command	ahm 재시작 명령어
restore_command	아카이브 복구 명령어
archive_cleanup_command	재시작 지점 후 정리 명령어
use_recovery_min_apply_delay	복구 구성에서 recovery_min_apply_delay를 활성화/비활성화
recovery_min_apply_delay	복구 변경 사항을 적용하는 최소 지연 시간
db_connect_timeout	데이터베이스 서버에 대한 연결 시간 초과
primary_conninfo_options	primary_conninfo 복구에 대한 추가 옵션
standby_startup_delay	Standby 기동 대기 시간

use_pg_rewind	pg_rewind을 이용한 복구 설정
beacon_message_interval_seconds	노드 상태 확인 주기 (단위 : 초), AHM 노드 장애를 감지하기 위한 내부 상태 점검 메커니즘으로도 사용됩니다.
beacon_message_retry_count	장애 노드로 변경하기 위한 노드 상태 체크 실패 횟수
promote_leader_timeout_seconds	Standby에서 Primary 로 promote시키는 명령에 대한 타임아웃
wait_for_primary_timeout_seconds	시작 시 클러스터가 Primary 노드를 기다리는 대기 시간
pid_file_name	AHM PID 파일명(ahm.pid)
failover_consensus_with_half_votes	2노드 만으로 FailOver 활성화

19.4.2 Ahm Failover Interval

- AHM 장애시

항목	파라미터	설정 값 (디폴트)
AHM 상태 체크	beacon_message_interval_seconds	8초 주기
AHM 상태 체크 재시도	beacon_message_retry_count	2회 반복
Primary 기동 대기	wait_for_primary_timeout_seconds	20초
Failover Interval		약 36초 ~ 44초

- Database 장애시

항목	파라미터	설정 값 (디폴트)
Database 상태 체크	health_check_period	30초 주기
Database 응답대기	db_connect_timeout	5초
Database 상태 체크 재시도	max_consec_failures	5회
Primary 기동 대기	wait_for_primary_timeout_seconds	20초

Failover Interval	195초 ~ 230초
-------------------	-------------

- **Example:**

노드 별로 각각 진행

-ahm_config.yml 구성

```
> cat ahm_config.yml
is_witness: false
priority : 1
data_dir: '/tmp/' #directory to hold temporary ahm data and pid file
#log_severity: DEBUG

ahm_port: 9500
ipc_port: 9595
socket_dir: /tmp
#nodes_filename: " #default CWD/nodes.conf
node_name: AHM-NODE-1
conn_string: host=localhost port=5432 dbname=postgres user=agens

beacon_message_interval_seconds: 8
beacon_message_retry_count: 2
promote_leader_timeout_seconds:    20
wait_for_primary_timeout_seconds:  20

#
#   PG Database Health Check Configurations
#
db_connect_timeout: 5
health_check_period: 30
max_consec_failures: 5

#
#   Virtual IP configurations
#
virtual_IP: '210.104.181.42'
if_cmd_path: '/bin'
if_up_cmd: '/usr/bin/sudo /sbin/ip addr add $_IP_$ /24 dev eno1 label eno1:0'
if_down_cmd: '/usr/bin/sudo /sbin/ip addr del $_IP_$ /24 dev eno1'
arping_path: '/usr/sbin'
arping_cmd: '/usr/bin/sudo /usr/sbin/arping -U $_IP_$ -w 1 -I eno1'
ping_path : '/bin'
```

```
#
#   Logging configurations
#
log_filename: ""
log_prefix: '%l PID:%p %t '
log_verbose: false #true value

#
#   Database cluster configuration
#
db_user: 'agens'
db_database: 'postgres'
db_port: 5432
db_bin_dir: '/home/agens/AgensSQL-2.13.7.0/bin'
db_data_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'
db_config_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'

#
#   Promotion configurations
#
pre_promotion_command: '/home/agens/AHM/etc/failover.sh'
promotion_command: ""
post_promotion_command: ""

#
#   Failover configurations
#
#follow_command:"
standby_startup_delay: 3
use_pg_rewind: false
failover_consensus_with_half_votes: true

#
#   Replication configurations
#
restore_command: ""
archive_cleanup_command: ""
use_recovery_min_apply_delay: false
recovery_min_apply_delay: 0
primary_conninfo_options: ""

pid_file_name : 'ahm.pid'
```


-Failover Script 구성

```
> cat ahm_config.yml
is_witness: false
priority : 1
data_dir: '/tmp/' #directory to hold temporary ahm data and pid file
#log_severity: DEBUG

ahm_port: 9500
ipc_port: 9595
socket_dir: /tmp
#nodes_filename: " #default CWD/nodes.conf
node_name: AHM-NODE-1
conn_string: host=localhost port=5432 dbname=postgres user=agens

beacon_message_interval_seconds: 8
beacon_message_retry_count: 2
promote_leader_timeout_seconds:    20
wait_for_primary_timeout_seconds:  20

#
#   PG Database Health Check Configurations
#
db_connect_timeout: 5
health_check_period: 30
max_consec_failures: 5

#
#   Virtual IP configurations
#
virtual_IP: '210.104.181.42'
if_cmd_path: '/bin'
if_up_cmd: '/usr/bin/sudo /sbin/ip addr add $_IP_$ /24 dev eno1 label eno1:0'
if_down_cmd: '/usr/bin/sudo /sbin/ip addr del $_IP_$ /24 dev eno1'
arping_path: '/usr/sbin'
arping_cmd: '/usr/bin/sudo /usr/sbin/arping -U $_IP_$ -w 1 -I eno1'
ping_path : '/bin'

#
#   Logging configurations
```

```
#
log_filename: ''
log_prefix: '%l PID:%p %t '
log_verbose: false #true value

#
#   Database cluster configuration
#
db_user: 'agens'
db_database: 'postgres'
db_port: 5432
db_bin_dir: '/home/agens/AgensSQL-2.13.7.0/bin'
db_data_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'
db_config_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'

#
#   Promotion configurations
#
pre_promotion_command: '/home/agens/AHM/etc/failover.sh'
promotion_command: ''
post_promotion_command: ''

#
#   Failover configurations
#
#follow_command:''
standby_startup_delay: 3
use_pg_rewind: false
failover_consensus_with_half_votes: true

#
#   Replication configurations
#
restore_command: ''
archive_cleanup_command: ''
use_recovery_min_apply_delay: false
recovery_min_apply_delay: 0
primary_conninfo_options: ''

pid_file_name : 'ahm.pid'
```

- **Ahm** 기동

```
> ahm -f /home/agens/etc/ahm_config.yml
```

19.4.3 노드 구성

노드 구성 파일에는 클러스터에 있는 모든 노드의 주소가 포함되어 있습니다.

파일은 IP:PORT 형식을 사용하여 AHM 노드 주소를 나타내고 각 라인에 정확히 하나의 IP:PORT 항목이 들어 갑니다.

예를 들어, 2개의 노드 AHM 클러스터의 nodes.conf 파일은 다음과 같습니다.

```
> cat nodes.conf
# IP:port list of AHM nodes
# one ip:port pair per line
# empty and commented lines will be ignored

210.104.181.77:9500
210.104.181.78:9500
```

기본적으로 ahm은 현재 작업 디렉토리에서 nodes.conf 파일을 찾고, nodes.conf의 경로는 **nodes_filename** 구성 매개변수를 사용하여 ahm 구성 파일에서 구성할 수도 있습니다.

Note

모든 AHM 노드에 대해 동일한 nodes.conf 파일을 사용해도 됩니다. AHM은 로컬 노드에 대한 항목을 자동으로 감지하고 원격 노드 주소만 사용하여 클러스터를 생성합니다.

19.5 AHM 실행

AHM을 실행하려면 ahm binary를 실행하세요.

```
ahm [ -f CONFIG_FILE ] [ -d ] [ -n ] [stop|start]
```

Example:

```
> ahm -f /home/agens/etc/ahm_config.yml
```

AHM은 다양한 명령줄 인수를 허용합니다. 허용되는 인수 목록을 `ahm -help`을 사용하여 확인 할 수 있습니다.

```
> ahm -- help
ahm version 1.0,
HA manager for Agens[SQL|Graph] server

Usage:
ahm [ -f CONFIG_FILE ] [ -d ] [stop|start]
-f, --config-file=CONFIG_FILE
    Set the path to the ahm_config.yml configuration file
    (default: /usr/local/etc/ahm_config.yml)
-h, --help      Print this help
-d, --debug     Debug mode
-n --no-detach  Do not detach terminal
```

Note

모든 AHM 노드에 대해 동일한 `nodes.conf` 파일을 사용해도 됩니다. AHM은 로컬 노드에 대한 항목을 자동으로 감지하고 원격 노드 주소만 사용하여 클러스터를 생성합니다.¹¹

19.6 AHM 유틸리티 (Client Interface)

AHM은 DB 관리자를 위한 클러스터 관리 및 모니터링 인터페이스를 제공합니다. 이를 통해 클러스터의 상태 확인, 클러스터 종료, Standby 노드를 attach 또는 promote와 같은 작업을 수행할 수 있습니다.

19.6.1 ahm_cluster_status

본 유틸리티는 현재 클러스터 상태를 제공합니다. 해당 유틸리티를 다음 옵션과 함께 사용할 수 있습니다.

```
ahm_cluster_status [option ...]  
-h, --host= 대상서버 IP또는 VIP  
-p, --port= PORT
```

클러스터가 Primary 및 Standby 2개의 노드로 구성되어 있는 경우 다음과 같은 결과를 확인할 수 있습니다.

```
> ahm_cluster_status -p 9595 -h 210.104.181.42
```

Cluster Status

```
Leader Node : AHM-NODE-1  
Leader Host : 210.104.181.77  
Successor Node : AHM-NODE-2  
Successor Host : 210.104.181.78  
VIP : 210.104.181.42  
Leader-hold-VIP : True  
Remote Nodes : 1
```

Node	Database	Address	Port	Priority	Witness	State
AHM-NODE-1	PRIMARY	210.104.181.77	9500	1	False	LEADER
AHM-NODE-2	STANDBY	210.104.181.78	9500	2	False	STANDBY

--

다음은 AHM 노드의 State-name 입니다.

PG State Number	STATE-NAME	DEFINITION
0	DEAD	Node is not initialized
1	LOADING	Loading configuration files
2	INITIALIZING	
3	PARTICIPATE_IN_ELECTION	
4	STAND_FOR_LEADER	Node contested as Leader in election
5	LEADER	Node is cluster leader
6	STANDBY	Node is a standby
7	LOST	Node is Lost and not reachable
8	WAITING_FOR_LEADER_FORMATIO N	Node is waiting for Leader formation
9	TAKING_OVER	Node announcing a leader contestant
10	IN_RESOURCE_LIMITATION	Node does not have valid Standby to be a leader
11	SHUTDOWN	Node is shut downed

다음은 AHM 노드로 구성된 데이터베이스의 state-name 입니다.

PG State Number	STATE-NAME	DEFINITION
0	UNKNOWN	Unknown state
1	PRIMARY	Primary database
2	STANDBY	Standby database
3	FAILED	Failed database
4	QUARANTINED	Quarantined
5	NOT_CONFIGURED	Database not configured

19.6.2 ahm_attach_node

본 유틸리티는 이전에 연결이 끊긴 ahm 노드를 클러스터에 연결해줍니다.

ahm_attach_node [Option...]

-h, --host= 대상 서버 IP

-p, --port= PORT

```
> ahm_attach_node -h 210.104.181.78 -p 9595
```

Command Success

Info: attach database executed successfully

19.6.3 ahm_shutdown_cluster

ahm_shutdown_cluster 유틸리티는 AgensSQL 데이터베이스를 중단하지 않고 모든 AHM 노드를 종료합니다.

ahm_shutdown_cluster

```
> ahm_shutdown_cluster
```

Command Success

Info: Shutdown request sent to remote nodes

19.6.4 ahm_promote_node

본 유틸리티는 AgensSQL 서버를 새로운 Primary AgensSQL 서버로 promote 시킵니다.

ahm_promote_node [Option...]

-h, --host= 대상 서버 IP

-p, --port= PORT

```
> ahm_promote_node -h 210.104.181.77 -p 9595
```

Command Success

Info: Promote node task successfully scheduled

19.7 지원하는 **Failover** 시나리오

AHM은 클러스터로 구성된 노드의 장애를 모니터링하여 **failover**의 발생을 탐지합니다.

AHM의 **failover** 시나리오 세트는 매우 구체적이고 제한적입니다.

* **Failover** : Primary-Standby로 구성된 HA구성에서 Primary 노드가 장애가 발생하여 서비스를 못할 경우 HA솔루션에서 감지하여 Standby 서버가 서비스 가능한 Primary로 전환 되는 과정

- Failover 발생 할수 있는 상황
 - ➔ Primary database가 충돌하거나 종료된 경우
 - ➔ Primary database를 hosting하는 노드가 충돌하거나 연결할 수 없는 경우

AHM은 이러한 조건의 정확성을 확인하기 위해 모든 시도를 합니다. AHM이 기본 데이터베이스 또는 노드에 장애가 발생했음을 확인할 수 없는 경우 AHM은 클러스터에서 **Failover** 작업을 수행하지 않습니다.

AHM은 다음과 같은 **Failover** 시나리오를 지원합니다.

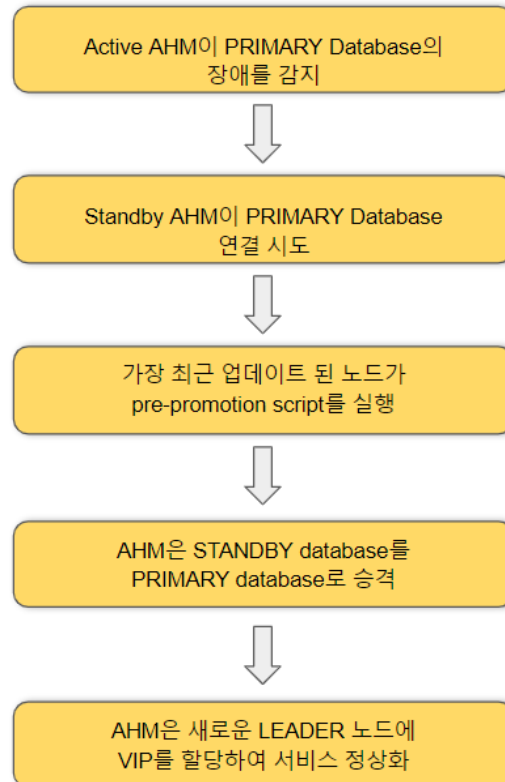
19.7.1 Primary database is down

Primary database 노드에서 실행 중인 AHM이 Primary database에서 발생한 장애를 감지하면 AHM은 장애 확인 프로세스를 시작합니다.

Primary 노드의 AHM이 Primary 데이터베이스에서 실패를 감지하면 모든 AHM이 Primary 데이터베이스에 직접 연결을 시도합니다.

Primary 데이터베이스 서버에 연결할 수 있는 AHM이 없는 경우 AHM은 **failover** 프로세스를 시작합니다.

최신 노드의 Standby AHM은 pre-promotion 스크립트(해당하는 경우)를 실행하고 Standby 데이터베이스를 Primary 데이터베이스로 promote 하고 virtual IP 주소를 할당합니다.



- 장애 노드 복구 방법

1. 장애 발생한 노드에서 데이터베이스를 다시 복구 해주세요.
2. 장애를 복구한 노드에서 `ahm_attach_node` 명령어를 호출합니다.

- 복구 된 Standby Node를 Primary 로 재전환

장애가 발생하였던 노드를 Standby 로 복구 후 Primary로 다시 변환 할 수 있습니다.
복구 된 Node에서 `ahm_promote_node` 명령어를 호출하여 노드를 Primary 역할로 다시 승격합니다.

Note

AHM은 장애가 발생한 Primary 데이터베이스를 재구축하여 Standby 상태로 만들지

않습니다. 재구성하기 전에 **Primary** 가 실패한 이유를 확인하고 새 **Primary** 에서 모든 데이터를 사용할 수 있는지 확인하는 것이 중요합니다. 서버를 **Standby** 상태로 복원할 준비가 되면 이전 데이터 디렉토리를 제거하고 서버를 복원할 수 있습니다. 자세한 내용은 **Standby** 서버 설정에 대한 **AgensSQL** 설명서를 참조하십시오.

19.7.2 Standby database is down

Standby AHM이 데이터베이스 장애를 감지하면 다른 **AHM**에 알리고 노드를 **LOST** 상태로 표시합니다.

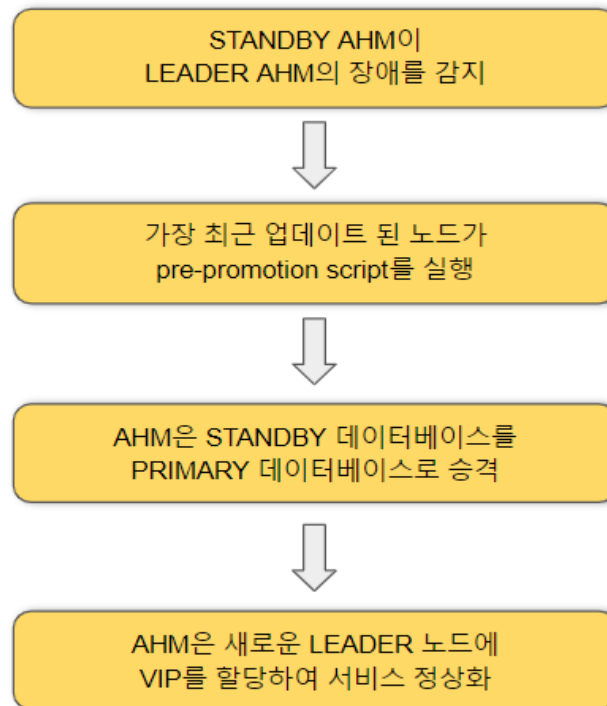
Standby 데이터베이스를 정상 상태로 되돌린 후 `ahm_attach_node` 명령을 호출하여 **standby** 데이터베이스를 클러스터로 되돌립니다.

19.7.3 Primary AHM exits or node fails

리더 노드가 충돌하거나 노드에 장애가 발생하면 **Standby** 노드는 장애를 감지하고 적절한 경우 장애 조치를 시작합니다.

Primary AHM이 클러스터에서 종료될 경우, 모든 **AHM**은 **Primary** 데이터베이스에 직접 연결하도록 합니다. 데이터베이스에 연결할 수 있으면, **AHM**은 **Primary AHM**의 장애를 통지합니다.

최신 노드의 **Standby AHM**은 **fencing** 스크립트(해당되는 경우)를 실행하고 **Standby** 데이터베이스를 **Primary** 데이터베이스로 **promote**하고 **virtual IP** 주소를 **standby** 노드에 할당합니다. 해당되는 경우 **AHM**은 **post-Promotion** 스크립트를 실행합니다.

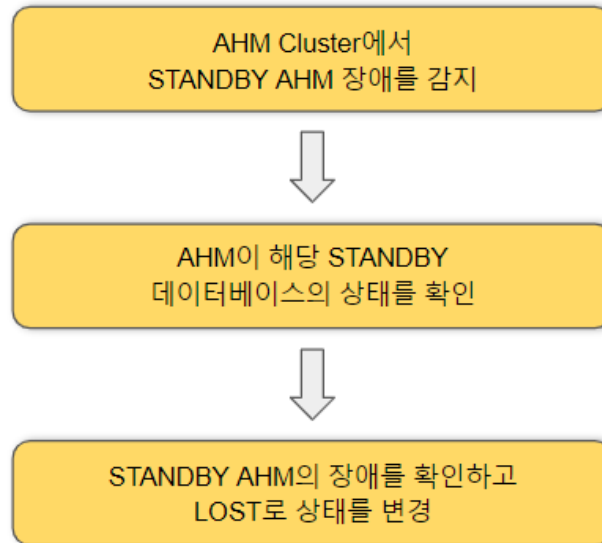


19.7.4 Standby AHM exits or node fails

클러스터에서 Standby AHM이 종료하거나 실패 오류가 발생하면, 다른 AHM은 해당 Standby AHM이 클러스터에 연결되지 않음을 감지합니다.

장애가 감지되면 이외의 AHM은 해당 STANDBY 데이터베이스에 연결을 시도합니다.

AHM이 문제가 있음을 확인하면 AHM은 해당 노드를 손실된 것으로 표시하고 LOST 상태로 변경합니다.



19.8 시나리오 Example

- **AHM 구성**
구성정보

Physical	Primary	Standby
IP	210.104.181.77	210.104.181.78
VIP	210.104.181.42	
Port	5432	
AGHOME	/home/agens/AgensSQL-2.13.7.0/	
PGDATA	/home/agens/AgensSQL-2.13.7.0/db_cluster	
Replication User	repluser	

- 1) AHM 설치 진행 (Primary-Standby 각 노드에 모두 구성)

AHM 설치파일 압축해제

```
$ tar -zxvf AHM_v1.0.tar.gze
```

- 2) 환경 설정 (Primary-Standby 각 노드에 모두 구성)

```
$ vi ~/.bash_profile
export PATH=$AGHOMe/bin:/home/agens/AHM/bin:$PATH
source ~/.bash_profile
```

agens User sudo 권한 추가 (root User로 진행)

```
$ vi /etc/sudoers
## Allows people in group wheel to run all commands
%wheel ALL=(ALL)    ALL
agens ALL=(ALL)      ALL

## Same thing without a password
agens                ALL=(ALL)    NOPASSWD: ALL
```

각 노드 간 SSH 설정 진행

ssh 설정

```
(root User 진행)
ssh-keygen -t rsa
ssh-copy-id root@210.104.181.77
ssh-copy-id root@210.104.181.78
ssh-copy-id agens@210.104.181.77
ssh-copy-id agens@210.104.181.78
```

```
(agens User 진행)
ssh-keygen -t rsa
ssh-copy-id root@210.104.181.77
ssh-copy-id root@210.104.181.78
ssh-copy-id agens@210.104.181.77
ssh-copy-id agens@210.104.181.78
```

3) AHM Config 설정 (Primary-Standby 각 노드에 모두 설정)

AHM Config

```
$ cp /home/agens/AHM/etc/ahm_config.yml.sample /home/agens/AHM/etc/ahm_config.yml
```

node_name 부분 이외 모두 동일하게 구성

```
$ vi /home/agens/AHM/etc/ahm_config.yml
is_witness: false
priority : 1
data_dir: '/tmp/' #directory to hold temporary ahm data and pid file
#log_severity: DEBUG

ahm_port: 9500
ipc_port: 9595
socket_dir: /tmp
#nodes_filename: " #default CWD/nodes.conf
node_name: AHM-NODE-1
conn_string: host=localhost port=5432 dbname=postgres user=agens

beacon_message_interval_seconds: 8
beacon_message_retry_count: 2
promote_leader_timeout_seconds:    20
wait_for_primary_timeout_seconds:  20

#
#   PG Database Health Check Configurations
#
db_connect_timeout: 5
health_check_period: 30
max_consec_failures: 5

#
#   Virtual IP configurations
#
virtual_IP: '210.104.181.42'
if_cmd_path: '/bin'
if_up_cmd: '/usr/bin/sudo /sbin/ip addr add $_IP_$/24 dev eno1 label eno1:0'
if_down_cmd: '/usr/bin/sudo /sbin/ip addr del $_IP_$/24 dev eno1'
arping_path: '/usr/sbin'
arping_cmd: '/usr/bin/sudo /usr/sbin/arping -U $_IP_ $ -w 1 -I eno1'
ping_path : '/bin'

#
#   Logging configurations
#
log_filename: ""
log_prefix: '%l PID:%p %t '
log_verbose: false #true value

#
#   Database cluster configuration
```

```
#
db_user: 'agens'
db_database: 'postgres'
db_port: 5432
db_bin_dir: '/home/agens/AgensSQL-2.13.7.0/bin'
db_data_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'
db_config_dir: '/home/agens/AgensSQL-2.13.7.0/db_cluster'

#
#   Promotion configurations
#
pre_promotion_command: '/home/agens/AHM/etc/failover.sh'
promotion_command: ""
post_promotion_command: ""

#
#   Failover configurations
#
#follow_command:"
standby_startup_delay: 3
use_pg_rewind: false
failover_consensus_with_half_votes: true

#
#   Replication configurations
#
restore_command: ""
archive_cleanup_command: ""
use_recovery_min_apply_delay: false
recovery_min_apply_delay: 0
primary_conninfo_options: ""

pid_file_name : 'ahm.pid'
```

Nodes Config

```
$ cp nodes.conf.sample nodes.conf
```

```
$ vi nodes.conf
210.104.181.77:9500 #Node1 IP:AHM Port
210.104.181.78:9500 #Node2 IP:AHM Port
```

Fail-Over Script

```
$ cp failover.sh.sample failover.sh
```

```
$ vi failover.sh
HOST_IP=`ifconfig eno1 | grep 'inet ' | awk '{ print $2}'`
VIP=210.104.181.42

if [ "$HOST_IP" = 210.104.181.77 ]
then
    ssh -T root@210.104.181.78 '/sbin/ip addr del 210.104.181.42/24 dev eno1'
    ssh -T agens@210.104.181.78 '/home/agens/AgensSQL-2.13.7.0/bin/ag_ctl stop -D /home/agens/AgensSQL-2.13.7.0/db_cluster'
elif [ "$HOST_IP" = 210.104.181.78 ]
then
    ssh -T root@210.104.181.77 '/sbin/ip addr del 210.104.181.42/24 dev eno1'
    ssh -T agens@210.104.181.77 '/home/agens/AgensSQL-2.13.7.0/bin/ag_ctl stop -D /home/agens/AgensSQL-2.13.7.0/db_cluster'
else
    echo $HOST_IP
    echo "Not Nodes"
fi
exit 0
```

4) AHM 기동 (각 노드에서 실행)

```
$ nohup ahm -f /home/agens/AHM/etc/ahm_config.yml &
```

5) 상태 확인

```
$ Ahm_cluster_status -h 210.104.181.42
```

정상적인 Primary-Standby 상태를 확인 할 수 있습니다.

```
Cluster Status
Leader Node: AHM-NODE-1
Leader Host: 210.104.181.77
Successor Node: AHM-NODE-2
Successor Host: 210.104.181.78
Virtual IP : 210.104.181.42
```


Leader-Hold-VIP: True
Remote Nodes: 1

Node	Database	Address	Port	Priority	Witness	State
AHM-NODE-1	PRIMARY	210.104.181.77	9500	1	False	LEADER
AHM-NODE-2	STANDBY	210.104.181.78	9500	2	False	STANDBY

- **Primary Database 장애 시 Standby /Failover 기능**

1) Primary Node DB Down

```
$ ag_ctl stop
```

2) FailOver 확인

```
$ ahm_cluster_status -h 210.104.181.42
```

1번 노드의 장애로 “FAILED” 상태로 변경 되며 2번 노드로 Failover 완료 후 "PRIMARY" 상태로 보여짐을 확인 할 수 있습니다.

Cluster Status

Leader Node: AHM-NODE-2
Leader Host: 210.104.181.78
Successor Node: AHM-NODE-2
Successor Host: 210.104.181.78
Virtual IP : 210.104.181.42
Leader-Hold-VIP: False
Remote Nodes: 1

Node	Database	Address	Port	Priority	Witness	State
AHM-NODE-1	FAILED	210.104.181.77	9500	1	False	STANDBY
AHM-NODE-2	PRIMARY	210.104.181.78	9500	2	False	LEADER

- **Failover 노드 정상화**

1) FAILED Node 정상화 (FAILED Node에서 진행)

```
$ ag_ctl start
```

(DB 손상시 pg_basebackup를 이용하여 DB 복구 진행 필요)

2) AHM Attach 진행

```
$ ahm_attach_node
```

Command Success

Info: attach database executed successfully

3) 상태 확인

```
$ ahm_cluster_status -h 210.104.181.42
```

1번 노드의 장애로 “FAILED” 상태에서 정상화되면서 “STANDBY” 상태로 보여짐을 확인 할 수 있습니다.

Command Success

Info: attach database executed successfully

[agens:/home/agens]#ahm_cluster_status

Cluster Status

Leader Node: AHM-NODE-2

Leader Host: 210.104.181.78

Successor Node: AHM-NODE-2

Successor Host: 210.104.181.78

Virtual IP : 210.104.181.42

Leader-Hold-VIP: False

Remote Nodes: 1

Node	Database	Address	Port	Priority	Witness	State
AHM-NODE-1	STANDBY	210.104.181.77	9500	1	False	STANDBY
AHM-NODE-2	PRIMARY	210.104.181.78	9500	2	False	LEADER

- **Primary AHM 장애 시 Standby /Failover** 기능
- Primary Node AHM Down

```
$ ahm stop
```

- FailOver 확인

```
$ ahm_cluster_status -h 210.104.181.42
```

2번 노드의 장애로 State "LOST" 상태로 변경되며 2번 노드로 Failover 완료 후 State "LEADER" 상태로 보여짐을 확인 할 수 있습니다. (AHM이 LOST 상태에서는 Database State는 의미 없습니다.

```
Cluster Status
Leader Node: AHM-NODE-1
Leader Host: 210.104.181.77
Successor Node: Not Set
Successor Host: Not Set
Virtual IP : 210.104.181.42
Leader-Hold-VIP: False
Remote Nodes: 1

Node Database Address Port Priority Witness State
AHM-NODE-1 PRIMARY 210.104.181.77 9500 1 False LEADER
AHM-NODE-2 PRIMARY 210.104.181.78 9500 1 False LOST
```

- Failover 노드 정상화

4) LOST Node DB 정상화 (LOST Node에서 진행)

pg_basebackup를 이용하여 DB 복구 진행

```
$ rm -rf $PGDATA
$ pg_basebackup -h 210.104.181.77 -D $AGDATA -U repluser -p 5432 -v -P -R
--wal-method=stream
$ ag_ctl start
```

5) AHM 기동 진행 (LOST Node에서 진행)

```
$ nohup ahm -f /home/agens/AHM/etc/ahm_config.yml &
```

6) 상태 확인

```
$ ahm_cluster_status -h 210.104.181.42
```

2번 노드의 장애로 “LOST” 상태에서 정상화되면서 “STANDBY” 상태로 보여짐을 확인 할 수 있습니다.

Cluster Status

Leader Node: AHM-NODE-1

Leader Host: 210.104.181.77

Successor Node: AHM-NODE-2

Successor Host: 210.104.181.78

Virtual IP : 210.104.181.42

Leader-Hold-VIP: False

Remote Nodes: 1

Node	Database	Address	Port	Priority	Witness	State
AHM-NODE-1	PRIMARY	210.104.181.77	9500	1	False	LEADER
AHM-NODE-2	STANDBY	210.104.181.78	9500	1	False	STANDBY