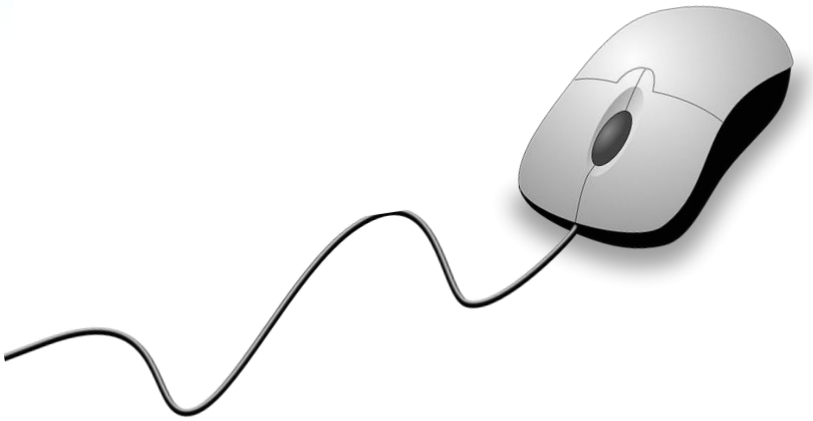
공개SW 솔루션 설치 & 활용 가이드

시스템SW -> 데이터관리



EDB™
POWER TO POSTGRES

제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

OPENUP
오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center

CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 아키텍처 및 기능 소개

1. 개요

소개	• 가장 완성도 높은 안정적인 오픈소스 DBMS PostgreSQL 기반 제품 • 오라클과 95% 이상 호환되는 유일한 오픈소스 DBMS		
주요기능	• Database Migration • Full ACID 호환 • 파티셔닝, 병렬처리		
대분류	• 시스템 SW	**소분류**	• 데이터관리
라이선스형태	• PostgreSQL 라이선스	**사전설치 솔루션**	• N/A
		버전	• EDB Postgres Advanced Server v13
특징	• 오라클과 95% 이상 호환되며 스키마 자동전환 툴 제공 • 클라우드 컨테이너 환경 구축 및 강력한 고가용성 기능 제공 • Java, python, perl 등 대부분의 언어를 API 형태로 제공		
개발회사/커뮤니티	• EnterpriseDB/PostgreSQL Global Development Group		
공식 홈페이지	• http://www.enterprisedb.com		

2. 기능요약

- EDB Postgres Advanced Server의 주요 기능

주요 기능	지원 여부
Full ACID 호환	지원
Table 파티셔닝	지원(Range, Hash, List, Composite)
Row Level Locking	지원
병렬처리	지원
MVCC	지원
온라인 백업	지원
Point-In-Time Recovery	지원
오라클 호환성	지원

3. 실행환경

- OS/Hardware/CPU/Database 제한

구분	EDB Postgres Advanced Server
지원되는 OS	RHEL 8.x / 7.x, CentOS 8.x / 7.x, OL 8.x / 7.x, Ubuntu 20.04, 18.04 LTS, Debian 10.x / 9.x Windows Server 2019, Windows server 2016
지원되는 CPU	x86_64, PowerPC
최소 사양	1 Ghz CPU processor 램 1GB 이상의 하드웨어 512MB 이상의 디스크 공간
Database Limitations	최대 Database Size: 무제한 최대 Table Size: 32 TB 최대 Row Size: 1.6 TB 최대 Column Size: 1 GB 테이블당 인덱스 개수: 무제한 테이블당 컬럼 개수: 250~1600 (컬럼 타입에 따라서 다름) 테이블당 최대 Rows: 무제한

4. 설치 및 실행

4.1 설치 준비

4.2 저장소 설치

4.3 저장소 패스워드 적용

4.4 패키지 설치

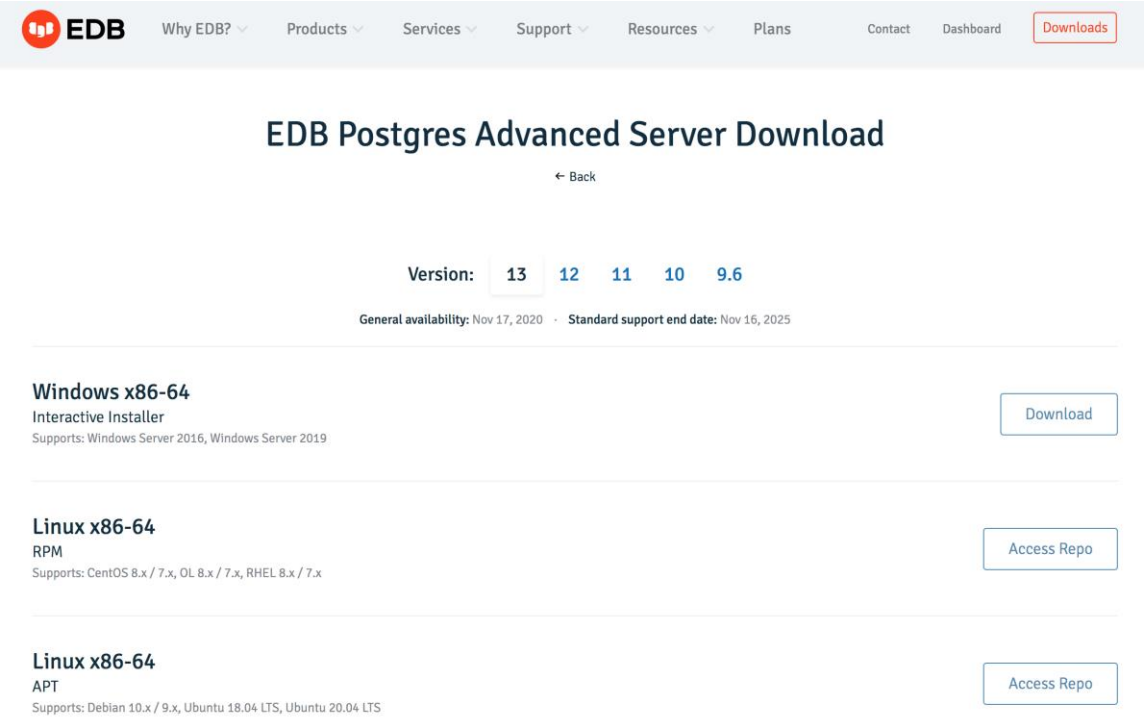
4.5 설치 후 과정

4.6 기동 및 종료

4. 설치 및 실행

4.1 설치 준비

- https://www.enterprisedb.com/downloads/edb-postgres-advanced-server / 에서 사용하는 리눅스 배포판과 버전 확인 및 EDB PostgreSQL 확인
- 버전 11이후에서는 설치용 패키지를 제공하지 않고, YUM Repository로 변경됨.
- OS 버전(RHEL 7/8, CENTOS)과 종류(RHEL/UBUNTU 등)에 따라 설치 및 환경 설정 방법이 다름



4. 설치 및 실행

4.2 저장소 설치

- RHEL(Redhat)과 CentOS는 커널이 같아서 숫자만 같으면 동일 rpm으로 설치 가능
- edb-repo 설치

RHEL 7/ CentOS 7

```
#> yum install -y https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

RHEL 8/ CentOS 8

```
#> dnf -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

Windows x86-64

- https://www.enterprisedb.com/downloads/edb-postgres-advanced-server
- Download를 클릭하여 인스톨러 버전 다운로드 후 GUI환경으로 설치

4. 설치 및 실행

4.3 저장소 패스워드 적용

- 사용자설정에 따라 다르지만보통 Linux의 경우/etc/yum.repos.d 디렉토리에 저장소(yum repository)를 구축
- edb-repo 접근을 의한 개정 및 패스워드 입력 (회원가입 필요)
- epel-release를 반드시 설치

```
#> sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo

#> yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

4. 설치 및 실행

4.4 패키지 설치

- yum install 명령으로 설치

#> yum -y install edb-as13-server

```
[root@kdshost ~]# yum -y install edb-as13-server
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
epel/x86_64/metalink                                                     | 6.6 kB  00:00:00
 * base: ftp.jaist.ac.jp
 * epel: mirror.yer.az
 * extras: ftp.jaist.ac.jp
 * updates: mirrors.cat.net
2ndquadrant-products-2ndqpostgres-release-pg12                           | 2.9 kB  00:00:00
2ndquadrant-products-2ndqpostgres-release-pg12-debug                     | 2.9 kB  00:00:00
2ndquadrant-products-tpa-release-pg12                                    | 2.9 kB  00:00:00
2ndquadrant-products-tpa-release-pg12-debug                              | 2.9 kB  00:00:00
base                                                                     | 3.6 kB  00:00:00
edb/7/x86_64/signature                                                   |  490 B  00:00:00
Retrieving key from file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
Importing GPG key 0x7E30651C:
 Userid     : "EnterpriseDB Inc. (EnterpriseDB Yum Repositories) <packages@enterprisedb.com>"
 Fingerprint: ca40 9f7c 635f 2ae5 6c9e 8b34 e5ed e919 7e30 651c
 Package    : edb-repo-20-3.noarch (@/edb-repo-latest.noarch)
 From       : /etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
edb/7/x86_64/signature                                                   | 2.5 kB  00:00:00 !!!
epel                                                                     | 4.7 kB  00:00:00
extras                                                                   | 2.9 kB  00:00:00
updates                                                                  | 2.9 kB  00:00:00
(1/7): 2ndquadrant-products-2ndqpostgres-release-pg12-debug/7/x86_64/primary_db | 9.7 kB  00:00:01
(2/7): 2ndquadrant-products-tpa-release-pg12/7/x86_64/primary_db         |  11 kB  00:00:01
(3/7): updates/7/x86_64/primary_db                                       | 9.5 MB  00:00:01
(4/7): edb/7/x86_64/primary_db                                           | 573 kB  00:00:03
(5/7): epel/x86_64/updateinfo                                            | 1.0 MB  00:00:07
(6/7): 2ndquadrant-products-2ndqpostgres-release-pg12/7/x86_64/primary_db |  53 kB  00:00:08
(7/7): epel/x86_64/primary_db                                            | 6.9 MB  00:05:15
Resolving Dependencies
--> Running transaction check
---> Package edb-as13-server.x86_64 0:13.3.7-1.rhel7 will be installed
--> Processing Dependency: edb-as13-server-sqlprotect(x86-64) = 13.3.7-1.rhel7 for package: edb-as13-server-13.3.7-
1.rhel7.x86_64
--> Processing Dependency: edb-as13-server-pltcl(x86-64) = 13.3.7-1.rhel7 for package: edb-as13-server-13.3.7-1.rhe
l7.x86_64
```

4. 설치 및 실행

4.5 설치 후 과정

- initdb 명령을 사용하여 EDB Cluster 생성
- 서비스 구동 및 부팅 시 자동 구동 되도록 설정
- DB 생성 및 설정

  - psql 실행

```
#> PGSETUP_INITDB_OPTIONS="-E UTF-8 --lc-collate='C' --lc-ctype='C'" /usr/edb/as13/bin/edb-as-13-setup initdb
#> systemctl enable edb-as-13.service
#> systemctl start edb-as-13.service
#>  su – enterprisedb
#>  psql postgres
```

4. 설치 및 실행

4.6 기동 및 종료

- EDB Postgres의 기동, 종료, reload는 pg_ctl이라는 EDB PostgreSQL 내부 명령어 수행
- EDB Postgres 설치 시 환경 변수로 Database의 데이터위치는 지정하므로 –D 옵션 생략 가능

- 기동: EDB Postgres 프로세스 시작

- 종료: EDB Postgres 프로세스 종료

- 상태: EDB Postgres 프로세스의 구동여부 확인

- 재시작: EDB Postgres 프로세스를 종료 후 재시작

- reload: EDB Postgres 프로세스를 재시작 없이 반영할 수 있는 일부 파라미터의 환경을 DB에 즉각 적용

- # pg_ctl reload 혹은 postgres=# select pg_reload_conf(); ➔ pg_ctl 명령어나Postgres 내부함수이용

4. 설치 및 실행

4.6 기동 및 종료

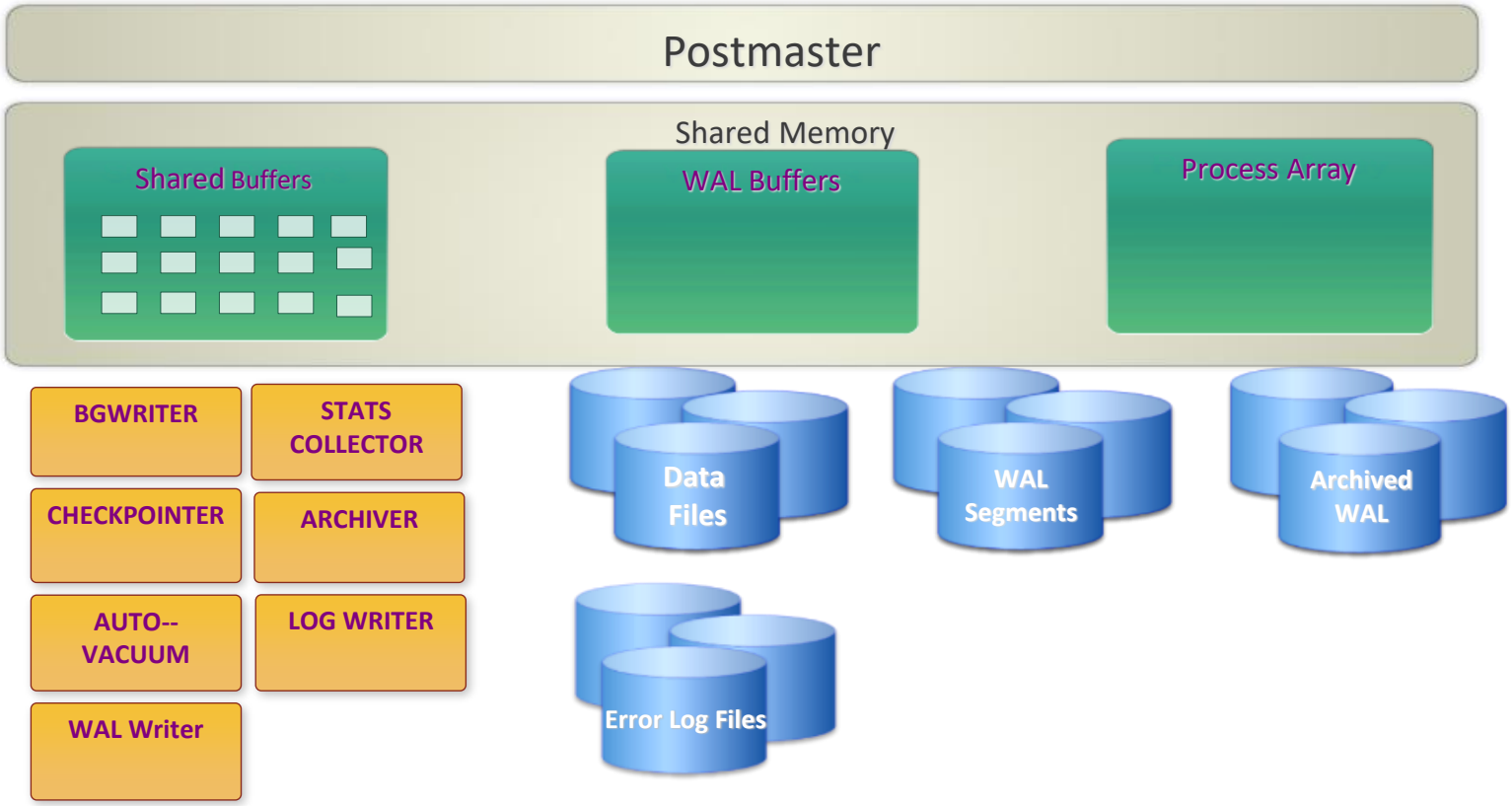
pg_ctl [start/stop/restart/reload/status]
pg_ctl start [-w] [-D DATADIR]
pg_ctl stop [-w] [-D DATADIR] [-m SHUTDOWN-MODE]
pg_ctl restart [-w] [-D DATADIR] [-m SHUTDOWN-MODE]
pg_ctl reload [-D DATADIR]
pg_ctl status [-D DATADIR] -D 옵션은 클러스터 위치 지정 -w 옵션은 명령이 정상적으로 수행될 때까지 기다림

Shutdown mode
smart : 모든 client들이 접속을 끊기를 기다림 (-ms), Oracle의 normal옵션과 비슷 (-ms)
fast : 모든 client들의 작업들을 rollback 하고 접속을 끊은후 종료하며, Oracle의 immediate과 비슷 (-mf)
immediate : 강제 종료하며 서버 시작 시 recovery 수행. Oracle의 abort옵션과 비슷 (-mi)
ex) pg_ctl stop –D /dbdata –mf –w

5. 아키텍처 및 기능 소개

5.1 아키텍처

- Process & Memory Architecture



5. 아키텍처 및 기능 소개

5.1 아키텍처

- 기본 Process 역할

  - Background writer : 변경 버퍼의 데이터를 디스크에 기록
  - WAL writer : WAL 버퍼의 데이터를 디스크에 기록
  - Checkpointer process : 파라미터 설정에 따라 checkpoint 수행
  - Autovacuum launcher : 필요에 따라 Autovacuum work 시작
  - Logging collector : 로그 내역 기록
  - Stats collector : 테이블과 블록 별 통계 수집
  - Archiver : 복구 작업을 위한 WAL 로그 파일 복사

5. 아키텍처 및 기능 소개

5.2 오라클 호환성 기능

- EDB Postgres는 Oracle로 부터의 응용 전환에 노력을 최소화 할 수 있도록 Oracle 호환 기능을 제공합니다.
- 데이터 타입에서 부터 주요 구문까지 Oracle의 구문을 그대로 사용할 수 있도록 함으로써 변환에 필요한 시간을 혁신적으로 단축 시킬 수 있습니다.
- 주요 기능
  - Data Type
    - Integer, number, char, double precision, float, varchar2, blob, clob, xmltype, rowid
  - 주요 구문 호환성
    - DUAL table
    - NVL, DECODE, ADD_MONTHS, TRUNC, SYSDATE 등
  - Built-in Package
    - DBMS_ : SQL, MVIEW, LOB, LOCK, CRYPTO 등 15개
    - UTL_ : FILE, HTTP, URL, SMTP, ENCODE 등 7개
  - Oracle Dictionary:
    - All_, DBA_, USER_ views
  - Tools
    - EDB*SQL, EDB*Loader, EDB*Wrap
  - 그 외 기능
    - Packages, Functions, Triggers, Hints, Database Links, Hierarchical Queries, Synonyms, Sequences
  - 지속적으로 호환성 기능 추가 중

5. 아키텍처 및 기능 소개

5.3 백업 방법

- EDB는 다양한 물리 및 논리 백업 방법을 지원하며 온라인중에도 DB의 핫백업을 지원합니다.
- 백업 시 압축을 사용하여 백업 용량을 최소화할 수 있습니다.

논리 백업 방법

- pg_dump 유틸리티
  - 논리 백업 전용 유틸리티
  - 온라인중에 수행 가능
  - Data only, DDL only, 특정 스키마 only, 특정 테이블 only 등 다양한 백업 옵션 제공
  - 일반 텍스트, 바이너리 등 백업 포맷 지원
  - 압축 지원
  - 리모트 백업 지원
  - 여러 테이블 백업시 병렬 처리 지원

물리 백업 방법

- SQL Command
  - SQL client 상에서 BEGIN/END 백업 명령어를 실행하여 데이터/로그 파일 백업
- pg_basebackup 유틸리티
  - 물리 백업 전용 유틸리티
  - 로컬/리모트 상의 DB 백업 지원
  - 압축 지원 (0~9 단계 조정 가능)
- BART (Backup and Recovery Tools)
  - pg_basebackup 유틸리티를 기반으로 백업 수행
  - 로컬/리모트 상의 여러 DB의 카탈로그 기능
  - 자동 압축 기능
  - 백업본 유지 정책 및 중복 제거 관리
  - 증분 백업 지원

5. 아키텍처 및 기능 소개

5.4 복구 방법

- EDB는 파일 시스템 기반 구조로 백업 및 복구에 높은 유연성을 제공 합니다.
- 논리 백업과 물리 백업을 조합하여 온라인 복구를 지원 합니다.

논리 백업 복구

- pg_restore 유틸리티
  - 논리 백업 복구 전용 유틸리티
  - 온라인중에 수행 가능
  - Data only, DDL only, 특정 스키마 only, 특정 테이블 only 등 다양한 복구 옵션 제공
  - 일반 텍스트, 바이너리 등 백업 포맷 지원
  - 압축 지원
  - 리모트 백업 지원
  - 여러 테이블 백업시 병렬 처리 지원

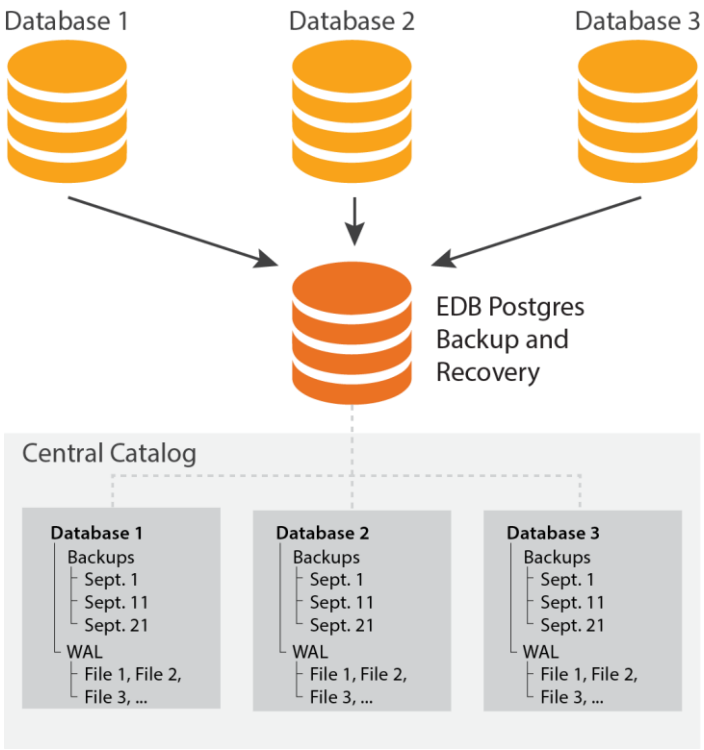
물리 백업 복구

- 완전 복구 지원
- 불완전 복구 지원
  - 백업 시 복구에 필요한 모든 아카이브 로그 자동으로 포함됨
- 시점 복구
  - 복구 설정 파일을 통한 절차 자동화
  - 시간 및 SCN 기준 복구 목표 설정 가능
  - 미리 정의한 복구 시점으로 복구 가능
- 복구 완료 후 수행할 작업 지정 가능
  - R/W 가능 상태로 오픈
  - 추가적인 복구를 위해 shutdown
  - Readonly 상태로 오픈
- BART (Backup and Recovery Tools)
  - 백업 카탈로그를 통한 복구 절차 자동화

5. 아키텍처 및 기능 소개

5.5 백업 도구

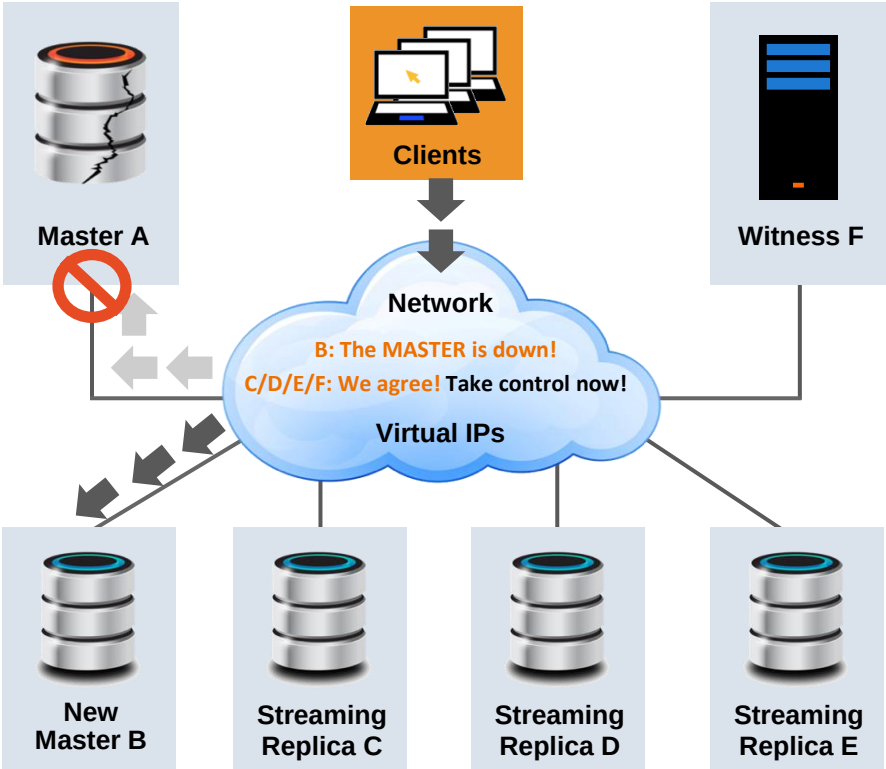
- BART (Backup and Recovery Tools)는 EDB Postgres의 백업/복구를 종합적으로 관리하는 EDB의 백업/복구 관리도구입니다.
- BART는 데이터베이스의 가용성을 증대시키고 백업/복구 작업의 복잡성을 제거합니다.
- BART의 주요 기능은 다음과 같습니다.
  - 로컬/리모트의 여러 PostgreSQL 또는 EDB Postgres 에 대한 카탈로깅을 통한 관리
  - 자동 압축 기능 및 MD5 checksum 검증 기능
  - 트랜잭션 로그 백업 및 시점 복구 지원
  - 테이블스페이스 복구 및 복구 위치 변경 지원
  - 백업본 유지 정책 및 중복 제거 관리



5. 아키텍처 및 기능 소개

5.6 EFM을 이용한 DB 이중화 구성의 예

- EDB Failover Manager - 자동 Failover 기능을 수행
  - 장애 감지 및 자동 Failover 지원 (on/off 가능)
  - Witness 에이전트를 통한 Split Brain 방지
  - VIP, Fencing Script 등을 통한 Fencing 설정
    a. fencing script
    b. promotion script
  - 2개 이상 다중 스탠바이 인스턴스 환경 지원
  - 자동 알림 기능 지원
  - 복제 상황 모니터링 지원
    a. 복제 환경 topology
    b. 복제 현황 모니터링
    c. 복제 lag 모니터링

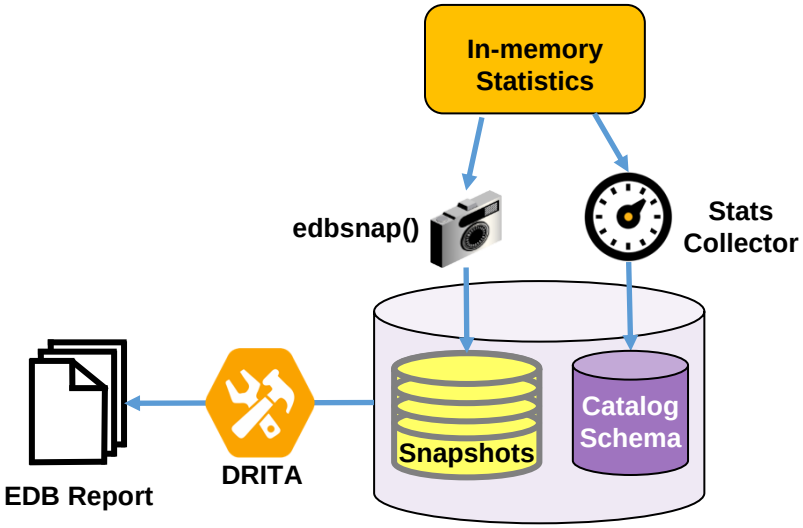


5. 아키텍처 및 기능 소개

5.7 성능 모니터링

- EPAS의 DRITA (Dynamic Runtime Instrumentation Tools Architecture) 기능은 개별 세션 및 데이터베이스에 발생하는 모든 대기 이벤트를 카탈로그 뷰에 스냅샷 단위로 저장합니다.
- 또한 Stats Collector에 의해 카탈로그 테이블에 실시간 수집되는 오브젝트에 엑세스 되는 모든 이벤트 정보와 결합하여 성능 통계 정보 보고서를 작성할 수 있습니다.
- EPAS의 DRITA 성능 통계 정보 보고서는 오라클의 Statspack, AWR 등과 유사한 정보들을 담고 있습니다.

보고서 함수명	설명
edbreport()	Database 보고서 조회
stat_tables_rpt()	테이블 보고서 조회
statio_tables_rpt()	테이블 IO 보고서 조회
stat_indexes_rpt()	인덱스 보고서 조회
statio_indexes_rpt()	인덱스 IO 보고서 조회
sys_rpt()	시스템 대기 이벤드 보고서 조회
sess_rpt()	세션 대기 이벤드 보고서 조회
edbsnap()	스냅샷 생성
get_snap()	스냅샷 목록 조회
purgesnap()	스냅샷 구간 삭제
truncsnap()	스냅샷 전체 삭제



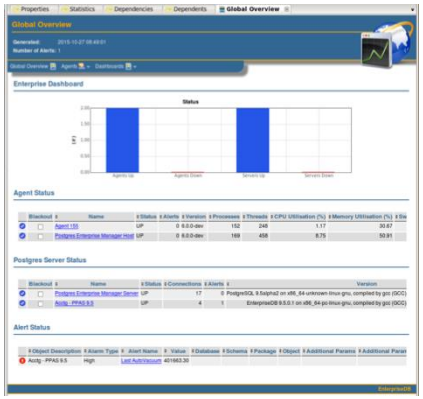
[그림 1] Dynamic Runtime Instrumentation Tools Architecture

5. 아키텍처 및 기능 소개

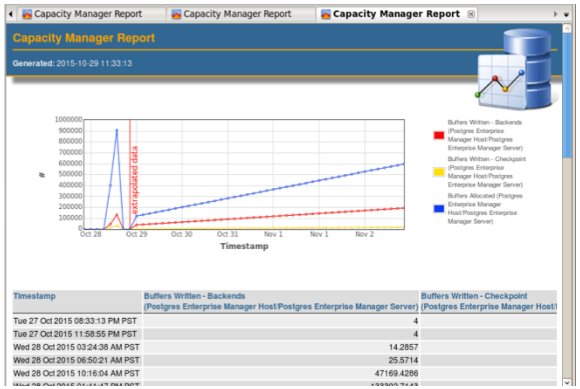
5.7 성능 모니터링 툴

- EPAS는 PEM (Postgres Enterprise Manager)를 통해 다양한 모니터링 정보와 알림 및 성능 개선을 위한 성능 분석 도구 및 튜닝 권고자 등 다양한 기능을 제공하고 있습니다.

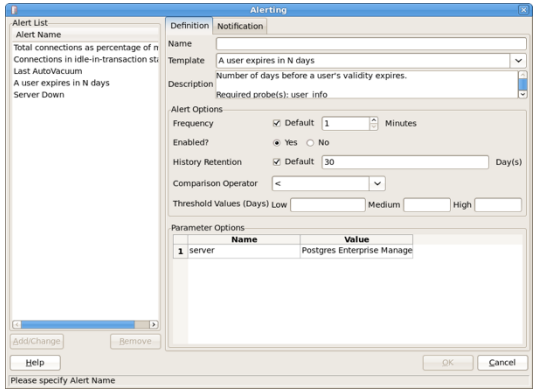
DB 성능 모니터링



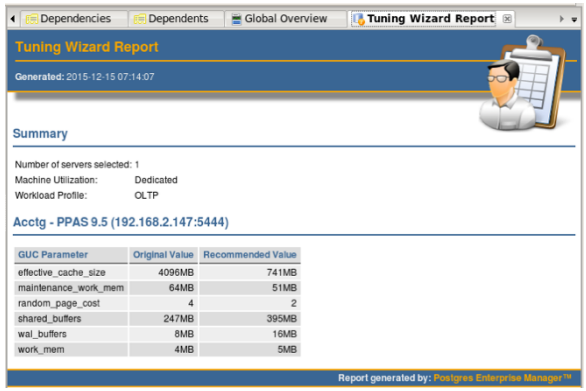
DB Capacity Manager



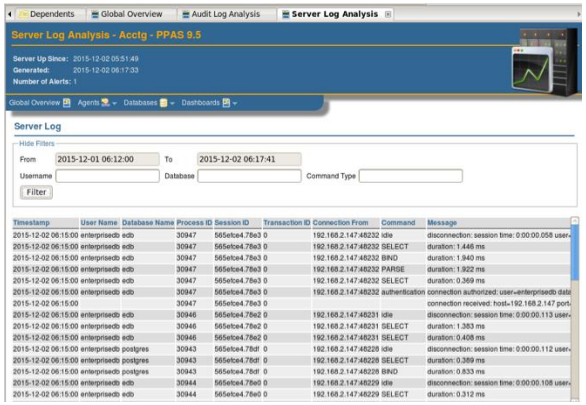
임계치에 의한 알림 설정



성능 분석 및 튜닝 권고자



로그 분석기



히스토그램 분석

- 컬럼 히스토그램은 pg_stats 카탈로그 뷰에 저장됩니다.
- 히스토그램 저장 컬럼 : elem_count_histogram
- ALTER TABLE table_name ALTER COLUMN column_name SET STATISTICS n 명령어로 히스토그램 저장 수준 조정 가능

Open Source Software Installation & Application Guide

이 저작물은 크리에이티브 커먼즈 [저작자표시 – 비영리 – 동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.