

5장. DB 환경 구성 파일 관리

총 4개의 환경 구성 파일이 존재합니다.

파일명	설명
postgresql.conf	기본 설정 파일(pg_settings Catalog View를 통해서도 확인 가능)
postgresql.auto.conf	ALTER SYSTEM에서 설정 한 구성 매개 변수를 저장하는 데 사용되며, Postgresql.conf 보다 postgresql.auto.conf 에 적용된 파라미터가 서버 재 시작 시 우선 적용됨
pg_hba.conf	PostgreSQL에 접속하는 클라이언트에 대한 인증 설정을 설명. HBA는 호스트 기반 인증(host-based authentication)의 약어
pg_ident.conf	클라이언트의 인증 방식으로 Ident 인증을 사용하는 경우, ident 사용자 이름을 PostgreSQL의 역할 이름에 매핑하는데 사용

5.1 postgresql.conf

주요 항목별 파라미터 설명은 아래와 같습니다.

5.1.1 Connection 설정

파라미터명	기본값	권장값	적용 유형	설명
listen_addresses	localhost	*	정적	접속 허용 address. '*'는 모든 IP 접속 허용. IP 접근 제어는 pg_hba.conf에서 적용 권장
port	5432	-	정적	기본적으로 서버가 listen하는 TCP 포트
max_connections	200	-	정적	동시에 접속할 수 있는 최대 수를 설정

5.1.2 Memory 설정

파라미터명	기본값	권장값	적용 유형	설명
shared_buffers	128MB	25~40%	정적	서버에서 사용할 공유 메모리 버퍼의 수를 설정합니다.
temp_buffers	8 MB		동적	각 세션에서 사용하는 임시 버퍼의 최대 수를

				설정합니다.
maintenance_work_mem	64 MB	Available Memory / 8	동적	유지 관리 작업에 사용할 최대 메모리를 설정합니다. vacuum, index 생성 등에서 일시적으로 사용됨
work_mem	4MB	(OS cache memory / connections) * 0.5	동적	쿼리 작업 공간에 사용할 최대 메모리를 설정합니다. 내부 정렬 및 해시 테이블에 사용할 메모리량 설정. 질의수행시에 사용하는 메모리. 정렬 속도 등에 영향을 줌.
effective_cache_size	-	Available Memory의 75%	동적	디스크 캐시 크기에 대한 planner가 작업할 메모리 설정

5.1.3 Log File 설정 및 관리

파라미터명	기본값	권장값	적용 유형	설명
log_directory	log	-	정적	로그 경로 지정
logging_collector	-	on	정적	stderr을 log파일로 저장
log_destination	stderr	-	정적	stderr 및 csvlog, syslog를 비롯한 서버 메시지를 로깅하는 몇 가지 메서드를 지원
log_filename	-	agensslog-%Y-%m-%d_%H%M%S.log	정적	로그 파일의 이름 패턴을 설정합니다.
log_file_mode	600		정적	Unix 시스템에서 매개 변수는 logging_collector가 사용 가능할 때 로그 파일에 대한 사용 권한 설정

5.1.4 Log File 내용 제어

파라미터명	기본값	권장값	적용 유형	설명
log_min_error_statement	error		동적	오류 조건을 발생시키는 SQL 문을 서버 로그에 기록 - DEBUG[1-5] 개발자를 위한 상세 정보 제공 - INFO 사용자가 암시적으로 요청한 정보 제공 - NOTICE user에게 도움이 될 수 있는 정보 전달 제공 - WARNING 경고에 해당하는 정보 제공

				- ERROR 현재 명령 중단의 원인이 되는 에러 정보 제공 - LOG 체크 포인트 활동과 같은 관리자에게 필요한 정보 제공 - FATAL 현재 세션을 중단의 원인이 되는 에러 정보 제공 - PANIC 모든 Database 세션 중단의 원인이 되는 에러 정보 제공
log_min_duration_statement	-1		동적	완료된 각 명령문의 duration 시간을 기록 (default : -1) • -1 : 비활성화 • 0 : 모든 명령문 지속 시간 기록 • 250ms : 250ms 이상 실행되는 모든 SQL 문 기록(파라미터 설정 값보다 오래 실행되는 쿼리에 대해 log file에 기록)
log_connections	off		정적	서버로의 연결 시도 및 접속 성공에 대한 정보
log_disconnections	off		정적	접속해제 로그
client_min_messages	notice		동적	클라이언트에 보낼 메시지의 level 설정
log_min_messages	warning		동적	서버 로그에 기록되는 메시지의 level 설정
log_timezone		Asia/Seoul		
log_statement	none	all	동적	logging 할 SQL문 제어 (default : none) • none(off) • ddl : CREATE, ALTER, DROP • mod : ddl 및 INSERT, UPDATE, TRUNCATE, COPY FROM 등 • all : ALL
log_line_prefix	'	'<%m-%c(%p):%x>'	정적	로그 Prefix패턴 • %u : 접속유저명 • %d : 디비명 • %p : 프로세스 ID • %i : command tag • %c : session id • %l : session line number • %s : session start timestamp • %x : transaction id • %q : stop here in non-session processes • %% : '%' 입력시 • %r : 리모트접속자의 HostName (Resolve)또는 IP와 Port • %t : Unix TimeStamp (일반적으로 사람이 보기 쉬운(흔한) 형태의 시간

5.1.5 WAL 설정 관련

파라미터명	기본값	권장 값	적용 유형	설명
wal_buffers		-1	정적	WAL 기능을 위해 공유 메모리에서 사용할 디스크 페이지 버퍼 수를 설정합니다. (-1은 shared_buffers에 기반하여 설정됨)
wal_level	minimal		정적	WAL 파일에 기록 할 데이터 양. Slave 서버에서 읽기 전용 쿼리에 필요한 정보 추가함 - minimal : 기본값 - archive : WAL 아카이브에 필요한 로깅만 추가 - hot_standby : 대기 서버에서 읽기 전용 쿼리에 필요한 정보 추가 - replica : PG 9.6 이상부터 archive와 hot_standby 모드가 합쳐진 모드, 하위 호환을 위해 archive/hot_standby로 설정시 본 모드로 동작
wal_writer_delay			동적	WAL 플러시 사이에 WAL 작성기의 절전 시간을 지정합니다.
max_wal_size	1GB		정적	검사 점을 트리거하는 WAL 크기를 설정합니다. PostgreSQL 버전 9.6 이하의 경우 max_wal_size는 16MB 단위입니다. PostgreSQL 버전 10 이상의 경우 max_wal_size는 1MB 단위입니다.
min_wal_size			정적	WAL을 축소할 최소 크기를 설정합니다. PostgreSQL 버전 9.6 이하의 경우 min_wal_size는 16MB 단위입니다. PostgreSQL 버전 10 이상의 경우 min_wal_size는 1MB 단위입니다.
checkpoint_timeout	5min	20min	동적	자동 WAL 체크포인트 사이의 최대 시간을 설정합니다.
checkpoint_completion_target	0.5	0.9	동적	체크포인트 도중 변경된 버퍼 플러시에 사용된 시간으로 체크포인트 간격의 분수 값입니다.

5.1.6 Archive 설정

파라미터명	기본값	권장값	적용 유형	설명
archive_mode	off	on	정적	archive_command 설정에 의해 아카이브 저장소로 전달 되도록 설정. wal_level이 minimal로 설정 된 경우 archive_mode를 사용할 수 없다
archive_command		test ! -f /archive/%f && cp %p /archive/%f		완료된 WAL 파일 세그먼트를 아카이브하기 위해 실행하는 로컬 셸 명령 - string : %p 아카이브할 파일의 경로명으로 대체. %f 파일명으로만 대체
archive_timeout		1~2min		archive_command는 완료된 WAL 세그먼트를 호출 하므로 WAL 트래픽이 발생되지 않아 트랜잭션이 완료되는 시간과 아카이브 저장소에서 안전하게 기록되는 사이에 긴 지연시간이 발생할 수 있음. 데이터가 아카이브되지 않은 채로 방치되지 않기 주기적으로 전환되도록 archive_timeout을 설정

5.1.7 autovacuum 관련

파라미터명	기본값	권장값	적용 유형	설명
autovacuum	on		동적	autovacuum 프로세스 사용 여부
autovacuum_freeze_max_age	2억		정적	한 테이블에서 마지막 VACUUM 작업으로 pg_class.relFrozenxid 값이 지정된 뒤로, 이 설정값 만큼 더 커지면 VACUUM 작업을 합니다. 트랜잭션 ID 겹치는 문제를 막기위한 것입니다. 알아야할 점은 어떤 다른 상황에서 autovacuum 작업을 무시하고 있어도, 이 작업은 진행됩니다. 초기값은 2억입니다. 이 값은 서버가 시작될 때만 지정할 수 있지만, 각 테이블 단위로 pg_autovacuum 시스템 테이블의 값을 지정함으로 줄일 수는 있습니다.
autovacuum_max_workers			정적	autovacuum 작업자 프로세스를 동시에 실행할 수 있는 최대 수를 설정합니다.
autovacuum_analyze_threshold	0.1		동적	한 테이블에서 autovacuum_analyze_threshold 값이

analyze_scale_factor				초과되어 ANALYZE 작업을 시작하려고 할때, 추가적으로 테이블 크기의 변화량도 함께 고려합니다. 여기서 지정한 만큼 크기가 커졌을 때, 실재 작업을 진행합니다. 초기값은 0.1 (테이블 크기의 10%)입니다.
autovacuum_analyze_thres hold	250		동적	ANALYZE 작업을 시작할 최소 자료 작업 변경 수, 초기값은 250입니다. 즉, 하나의 테이블에서 250개의 튜플이 추가되었거나, 갱신되었거나, 삭제 되었다면, ANALYZE 작업을 합니다.
autovacuum_naptime	1min		동적	autovacuum 데몬이 한번 작업하고, 쉬는 시간을 지정합니다. 이 시간을 경과하면, autovacuum 데몬은 서버가 관리하는 데이터베이스의 모든 테이블을 하나씩 VACUUM 명령과 ANALYZE 명령을 실행할지 안할지를 검사합니다. 이 쉬는 시간은 초단위로 지정할 수 있습니다.초기값은 1분이다(1m).
autovacuum_vacuum_cost_delay	-1		동적	autovacuum 데몬의 초과 비용 사용시 멈추는 최대 시간(밀리세컨드)입니다. 이 값이 -1(초기값)이면, VACUUM 작업을 멈추지 않는다. 특별히 지정하지 않으면, vacuum_cost_delay 설정값이 사용될 것입니다.
autovacuum_vacuum_cost_limit	-1		동적	autovacuum 데몬이 사용할 수 있는 최대 비용을 지정합니다. 이 값이 -1(초기값)이면, 작업비용에 제한을 두지 않는다. 특별히 지정하지 않으면, vacuum_cost_limit 설정값이 사용될 것입니다.
autovacuum_vacuum_scale_factor	0.2		동적	한 테이블에서 autovacuum_vacuum_threshold 값이 초과되어 VACUUM 작업을 시작하려고 할때, 추가적으로 테이블 크기의 변화량도 함께 고려합니다. 여기서 지정한 만큼 크기가 커졌을 때, 실재 작업을 진행합니다. 초기값은 0.2 (테이블 크기의 20%)입니다.
autovacuum_vacuum_thres hold	500		동적	VACUUM 작업을 시작할 최소 자료 작업 변경 수, 초기값은 500입니다. 즉, 하나의 테이블에서 500개의 튜플이 갱신되었거나, 삭제 되었다면, VACUUM 작업을 합니다.

5.1.8 Extension Parameters 설정

파라미터명	기본값	권장값	적용 유형	설명
shared_preload_libraries	none	pg_stat_statements,pg_hint_plan,pg_prewarm'		서버 시작 시에 사전 로드할 하나 이상의 공유 라이브러리를 쉼표를 사용하여 지정합니다. 이 매개변수는 서버 시작 시에만 설정 가능합니다. 지정된

				라이브러리를 찾지 못하면 서버 시작이 실패합니다.
pg_stat_statements.max	1000		정적	처리할 수 있는 최대 쿼리 수를 지정. pg_stat_statements 뷰에 보일 최대 로우 수 pg_stat_statements.max * track_activity_query_size 바이트만큼의 공유 메모리를 추가로 필요
pg_stat_statements.track	top		정적	수집할 쿼리문의 사용빈도에 따른 쿼리문 사용 통계 정보 수집범위를 지정. 슈퍼유저만 바꿀 수 있음 top : 자주 사용하는 쿼리 대상 all : 모든 쿼리 대상 none : 아무 쿼리도 수집하지 않음.
pg_stat_statements.track_utility (boolean)	on		정적	SELECT, INSERT, UPDATE, DELETE 구문 외 다른 구문들도 수집 대상으로 할 것인지를 지정 (on / off)
pg_stat_statements.save (boolean)	on		정적	서버가 중지 되고, 재실행 되었을 때, 마지막 중지 시점의 쿼리문 통계 정보를 저장할 것인지를 지정
pg_hint_plan.enable_hint	on			True enables pg_hint_plan.
pg_hint_plan.enable_hint_table	on			True enables hinting by table.
pg_hint_plan.parse_messages	INFO			Specifies the log level of hint parse error. Valid values are error, warning, notice, info, log, debug.
pg_hint_plan.debug_print	off			Controls debug print and verbosity. Valid values are off, on, detailed, and verbose.
pg_hint_plan.message_level	INFO			Specifies message level of debug print. Valid values are error, warning, notice, info, log, debug.

5.1.9 통계정보 관련

파라미터명	기본값	권장값	적용 유형	설명
track_activity_query_size	1024		정적	pg_stat_activity.current_query에 예약되는 크기(바이트)를 설정합니다.

track_activities	on		동적	각 세션에서 현재 실행 중인 명령의 실행이 시작될 때 해당 명령에 대한 정보 수집을 활성화
track_counts	on		동적	데이터베이스 작업에 대한 통계 수집을 활성화
track_functions	none		동적	함수 호출 횟수 및 사용된 시간의 추적을 활성화합니다. 프로시저 언어 함수 all 만 추적하기 위해 pl 를 지정하면 SQL 및 C 언어 함수도 추적
track_io_timing	off		동적	데이터베이스 I/O 호출의 타이밍을 활성화합니다. 운영 체제에 현재 시간을 반복해서 쿼리함으로써 일부 플랫폼에서는 상당한 오버헤드가 발생되므로 이 매개변수는 기본적으로 off 이다
stats_temp_directory (string)	pg_stat_tm p		정적	임시 통계 데이터를 저장할 디렉토리를 설정합니다. 디렉터리에 대한 상대 경로이거나 절대 경로일 수 있습니다.
log_planner_stats (boolean)			동적	planner 성능 통계를 서버 로그에 기록합니다.

5.1.10 SQL 실행계획 관련

파라미터명	기본값	권장값	적용 유형	설명
auto_explain.log_min_ duration			동적	5초 이상 소요되는 sql 에 대해 쿼리 실행계획을 로그에 기록
auto_explain.log_anal yze			동적	쿼리 실행계획을 로그에 기록. 그런데 EXPLAIN ANALYZE 명령문을 기반으로 로그를 남기기 때문에, 롱쿼리를 다시 실행시킨다는 리스크가 있습니다.(만약 롱쿼리가 갑자기 몰리는 상황에서, 로그를 남기기 위해 EXPLAIN ANALYZE 쿼리가 날라간다면, 롱쿼리가 2배로 실행되어 문제를 더 키울 수 있습니다.) 그리고 단일 롱쿼리만 파악할 수 있기 때문에, 짧지만 여러번 호출되어 문제를 일으키는 쿼리를 알 수 없단 단점이 있습니다.

debug_pretty_print			동적	구문과 실행 계획 트리를 들여쓰기 하여 표시합니다.
debug_print_parse			동적	각 쿼리의 구문 분석 트리를 기록합니다.
debug_print_plan			동적	각 쿼리의 실행 계획을 기록합니다.
debug_print_rewritten			동적	각 쿼리에서 재작성된 구문 분석 트리를 기록합니다.
default_statistics_target			동적	기본 통계 대상을 설정합니다.
enable_bitmapscan			동적	planner가 비트맵 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_hashagg			동적	planner가 해시된 집계 계획을 사용할 수 있도록 활성화합니다.
enable_hashjoin			동적	planner가 해시 조인 계획을 사용할 수 있도록 활성화합니다.
enable_indexscan			동적	planner가 인덱스 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_material			동적	planner가 구체화를 사용할 수 있도록 활성화합니다.
enable_mergejoin			동적	planner가 병합 조인 계획을 사용할 수 있도록 활성화합니다.
enable_nestloop			동적	planner가 중첩 루프 조인 계획을 사용할 수 있도록 활성화합니다.
enable_seqscan			동적	planner가 순차적 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_sort			동적	planner가 명시적 정렬 단계를 사용할 수 있도록 활성화합니다.
enable_tidscan			동적	planner가 TID 스캔 계획을 사용할 수 있도록 활성화합니다.
constraint_exclusion			동적	planner가 제약 조건을 사용하여 쿼리를 최적화하도록 활성화합니다.
cpu_index_tuple_cost			동적	인덱스 스캔 중 각 인덱스 항목을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
cpu_operator_cost			동적	각 연산자 또는 함수 호출을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
cpu_tuple_cost			동적	각 튜플(행)을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
cursor_tuple_fraction			동적	planner가 예상하는 검색할 커서 행의 분수 값을 설정합니다.

random_page_cost			동적	비순차적으로 가져온 디스크 페이지에 대한 planner의 예상 코스트를 설정합니다.
------------------	--	--	----	--

5.2 pg_hba.conf 파일

클라이언트 인증 설정 파일인 `pg_hba.conf` 안의 각 레코드는 연결 형식, 클라이언트 IP 주소 범위, 데이터베이스 이름, 사용자 이름 및 이러한 매개 변수와 일치하는 연결에 사용되는 인증 방법을 지정합니다. 어떤 레코드도 일치하지 않으면 액세스가 거부됩니다.

레코드는 다음 7가지 형식 중 하나입니다.

```
local database user auth-method [auth-options]
host database user address auth-method [auth-options]
hostssl database user address auth-method [auth-options]
hostnossl database user address auth-method [auth-options]
host database user IP-address IP-mask auth-method [auth-options]
hostssl database user IP-address IP-mask auth-method [auth-options]
hostnossl database user IP-address IP-mask auth-method [auth-options]
```

필드의 의미는 다음과 같습니다.

- **local**

이 레코드는 Unix 도메인 소켓을 사용한 연결에 해당합니다. 유형의 레코드가 없으면 Unix 도메인 소켓 연결은 불가능합니다.

- **host**

이 레코드는 TCP/IP를 사용한 연결에 해당합니다. `host` 레코드는 SSL 연결 시도 혹은 비 SSL 연결 시도와 일치합니다.

- **hostssl**

이 레코드는 TCP/IP를 사용한 연결 시도와 일치하지만, SSL 암호화를 사용한 연결에만 해당됩니다. 이 옵션을 사용하려면 서버는 SSL 지원이 내장되어 있어야 합니다. 또한 SSL은 `ssl` 환경 설정 매개 변수를 설정함으로써 서버 시작 시에 활성화되어야 합니다.

- **hostnossl**

이 레코드 유형은 `hostssl`과는 반대로 동작합니다. `SSL`을 사용하지 않는 `TCP/IP` 상의 연결 시도에 대해서만 일치합니다.

- **database**

이 레코드는 데이터베이스 이름을 지정합니다.

- `all` 값은 모든 데이터베이스와 일치하도록 지정합니다.
- `sameuser` 값은 요청된 데이터베이스가 요청된 사용자와 이름이 동일한 경우에 레코드가 일치하도록 지정합니다.
- `samerole` 값은 요청된 사용자가 요청된 데이터베이스와 이름이 동일한 `role`의 멤버여야 하는지 지정합니다. `superuser`는 직접 혹은 간접적으로 `role`의 명시적인 멤버가 아닐 경우, 단지 `superuser`라는 이유로 `samerole`에 대한 `role`의 멤버로 간주되지 않습니다.
- `replication` 값은 복제 연결이 요청되는 경우 레코드가 일치하도록 지정합니다.(복제 연결은 특정 데이터베이스를 지정하지는 않습니다). 이 경우가 아니라면 특정 `AgensSQL` 데이터베이스의 이름으로 사용됩니다. 쉼표로 구분해서 데이터베이스 이름을 여러 개 쓸 수 있습니다. 데이터베이스 이름이 포함된 파일은 파일 이름 앞에 `@`를 붙여서 지정 가능합니다.

- **user**

이 레코드와 일치하는 데이터베이스 사용자 이름을 지정합니다. `all` 값은 모든 사용자와 일치하도록 지정합니다. 이 외에는, 특정한 데이터베이스 사용자의 이름이거나 앞에 `+`를 붙인 그룹 이름입니다. 이러한 이유로, `superuser`는 단지 `superuser`라는 이유 때문이 아니라, 직접 혹은 간접적으로 `role`의 명시적 멤버인 경우에만 `role` 멤버로 간주됩니다. 쉼표로 구분해서 사용자 이름을 여러 개 쓸 수 있습니다. 사용자 이름이 포함된 파일은 파일 이름 앞에 `@`를 붙여서 지정 가능합니다.

- **address**

이 레코드와 일치하는 클라이언트 머신 주소를 지정합니다. 이 필드는 호스트 이름, IP 주소 범위 또는 아래 설명된 특수 키워드 중 하나를 포함할 수 있습니다. IP 주소는 CIDR 마스크 길이의, 점으로 구분된 십진수(`dotted decimal`) 표준 표기법으로 지정됩니다. 마스크 길이는 일치해야 하는 클라이언트 IP 주소의 상위 비트 수를 나타냅니다. 이것의 오른쪽에 있는 비트는 주어진 IP 주소에서 0이어야 합니다. IP 주소 및 `/`, CIDR 마스크 길이 사이에 공백이 있으면 안됩니다.

이러한 방법으로 지정된 IP 주소 범위의 전형적인 예시는 단일 호스트의 경우 `172.20.143.89/32`, 소규모 네트워크의 경우 `172.20.143.0/24`, 대규모 네트워크의 경우 `10.6.0.0/16`일 수 있습니다. `0.0.0.0/0`은 모든 IPv4 주소를 나타내며 `::/0`은 모든 IPv6 주소를

나타냅니다. 단일 호스트를 지정하려면 IPv4의 경우 CIDR 마스크 32를 사용하고 IPv6의 경우 128을 사용해야 합니다. 네트워크 주소 끝에 0을 빠트리면 안 됩니다.

사용자는 아무 IP 주소나 일치하도록 all을 쓸 수도 있고, 서버의 자체 IP 주소만을 일치하도록 samehost를 쓸 수도 있고, 서버가 직접 연결되는 서브넷의 모든 주소와 일치하도록 samenet을 쓸 수도 있습니다.

- **IP-address, IP-mask**

이 필드는 CIDR-address 표기의 대안으로 사용될 수 있습니다. 마스크 길이를 지정하는 대신 실제 마스크가 쉼표로 구분하여 지정됩니다. 예를 들면, 255.0.0.0은 IPv4 CIDR 마스크 길이 8을 나타내고, 255.255.255.255는 CIDR 마스크 길이 32를 나타냅니다. 이 필드는 host 및 hostssl, hostnossl 레코드에 적용됩니다.

- **auth-method**

연결이 이 레코드와 일치할 때 사용하는 인증 방법을 지정합니다. 가능한 선택안은 다음과 같습니다.

- **trust**
무조건 연결을 허용합니다. 이 방법은 패스워드나 다른 인증 없이 임의의 데이터베이스 사용자로 로그인하여 누구나 데이터베이스 서버에 연결할 수 있습니다.
- **reject**
무조건 연결을 거부합니다. 이것은 그룹에서 특정 호스트를 “필터링”할 때 유용합니다. 예를 들면, reject 줄은 특정 호스트의 연결을 차단하고, 그 이후의 줄은 특정 네트워크의 남은 호스트들과의 연결을 허용합니다.
- **md5**
클라이언트가 인증을 위해 double-MD5-hashed 패스워드를 제공해야 합니다.
- **password**
클라이언트가 인증을 위해 암호화되지 않은 패스워드를 제공해야 합니다. 패스워드는 네트워크 상에서 일반 텍스트로 전송되므로 신뢰하지 않는 네트워크에서 이것을 사용하면 안 됩니다.
- **gss**
GSSAPI를 사용하여 사용자를 인증합니다. 이것은 TCP/IP 연결에서만 사용할 수 있습니다.

- **sspi**
SSPI를 사용하여 사용자를 인증합니다. 이것은 Windows에서만 사용할 수 있습니다.
- **ident**
클라이언트의 ident 서버에 접속함으로써 클라이언트의 운영 체제 사용자 이름을 획득하고, 요청된 데이터베이스 사용자 이름과 일치하는지 확인합니다. Ident 인증은 TCP/IP 연결에서만 사용할 수 있습니다. 로컬 연결에 대해 지정하는 경우 피어(peer) 인증이 대신 사용됩니다.
- **peer**
클라이언트의 운영 체제 사용자 이름을 운영 체제에서 획득하고, 요청된 데이터베이스 사용자 이름과 일치하는지 확인합니다. 이것은 로컬 연결에서만 사용할 수 있습니다.
- **ldap**
LDAP 서버를 사용하여 인증합니다.
- **radius**
RADIUS 서버를 사용하여 인증합니다.
- **cert**
SSL 클라이언트 인증을 사용하여 인증합니다.
- **pam**
운영 체제에서 제공하는 PAM(Pluggable Authentication Modules)을 사용하여 인증합니다.
- **auth-options**
auth-method 필드 이후에 인증 방법에 대한 옵션을 지정하는 name=value 형식의 필드가 있을 수 있습니다.

@ 구문이 포함된 파일은, 공백 또는 쉼표로 구분된 이름 목록으로 읽습니다. pg_hba.conf처럼 #으로 표시된 주석 및 중첩된 @ 구문이 허용됩니다. 파일 이름 뒤에 @가 나오는 것이 절대 경로가 아니면 참조 파일이 있는 디렉토리의 상대 경로로 취급됩니다.

pg_hba.conf 레코드는 각 연결 시도에 대해 순차적으로 검사되므로 레코드의 순서는 중요합니다. 일반적으로 초기 레코드는 연결 일치 매개 변수는 치밀하고, 인증 방법은 느슨한 반면, 후기 레코드는 일치 매개 변수는 느슨하고 인증 방법은 강력합니다. 예를 들면, 로컬 TCP/IP 연결에 대한 trust 인증을 사용하려고 하면서 원격 TCP/IP 연결을 할 수도 있습니다. 이런 경우 127.0.0.1로부터 연결을 위한 trust 인증을 지정한 레코드는 다양한 허용 클라이언트 IP 주소에 대해 패스워드 인증을 지원하는 레코드 이전에 나타납니다.

5.3 pg_ident.conf 파일

Ident 또는 GSSAPI 같은 외부 인증 시스템을 사용하는 경우, 연결을 시작하는 운영 체제 사용자의 이름은 연결해야 하는 데이터베이스 사용자 이름과 다를 수 있습니다. 이런 경우 사용자 이름 맵을 사용하여 운영 체제 사용자 이름과 데이터베이스 사용자 이름을 맵핑할 수 있습니다. 사용자 이름 맵핑을 사용하려면 `pg_hba.conf` 옵션 필드에서 `map=map-name`을 지정해야 합니다. 이 옵션은 외부 사용자 이름을 수신하는 모든 인증 방법에서 지원됩니다. 서로 다른 연결에 서로 다른 맵핑이 필요할 수 있으므로 연결별로 사용할 맵을 지정하기 위해 사용할 맵의 이름은 `pg_hba.conf`의 `map-name` 매개 변수에서 지정됩니다.

사용자 이름 맵은 `ident` 맵 파일에서 정의되며, 기본적으로 이름은 `pg_ident.conf`이며 데이터 디렉토리에 저장됩니다.

<code>map-name system-username database-username</code>

주석 및 공백은 `pg_hba.conf`에서와 동일하게 처리됩니다. `map-name`은 `pg_hba.conf`에서 이 맵핑을 참고하기 위해 사용되는 임의의 이름입니다. 나머지 2개의 필드는 운영 체제 사용자 이름과 데이터베이스 사용자 이름을 지정합니다. 동일한 `map-name`을 여러 번 사용해서 단일 맵 내에서 여러 사용자 맵핑을 지정할 수 있습니다.

주어진 한 명의 운영 체제 사용자가 몇 명의 데이터베이스 사용자에 대응하는지에 대해서는 아무런 제한이 없습니다.(그 반대도 마찬가지). 따라서, 맵의 항목은 사용자가 동일함을 의미한다기보다 “이 운영 체제 사용자는 이 데이터베이스 사용자로서 연결이 허용된다”로 생각되어야 합니다. 사용자가 연결 요청을 한 데이터베이스 사용자 이름을 사용하여 외부 인증 시스템에서 획득한 사용자 이름과 쌍을 이루는 맵 항목이 있을 경우 연결이 허용됩니다.

`system-username` 필드가 슬래시(/)로 시작되는 경우 필드의 나머지는 정규식으로 처리됩니다. 정규식은 단일 캡처 또는 괄호 표현식을 포함할 수 있으며, \1 (역슬래시)로 `database-username` 필드에서 참조가 가능합니다. 이것은 한 줄로 된 여러 사용자 이름을 맵핑할 수 있으며, 단순 구문 대체 시 특히 유용합니다. 예를 들면, 다음과 같습니다.

<code>mymap /^(.*)@mydomain\.com\$ \1</code> <code>mymap /^(.*)@otherdomain\.com\$ guest</code>
--

이 항목은 `@mydomain.com`로 끝나는 시스템 사용자 이름을 사용하여 사용자에게 대한 도메인 부분을 삭제하고, 시스템 이름이 `@otherdomain.com`로 끝나는 모든 사용자가 `guest`로 로그인하는 것을 허용합니다.