

목차

PostgreSQL 사용자 계정 접근설정	3
1. 파일의 주요 역할	3
2. 레코드(규칙)의 형식	3
3. 레코드의 필드 설명	3
3.1. 첫 번째 필드: 연결 유형 (Connection Type)	3
3.2. DATABASE 필드	4
3.3. USER 필드	4
3.4. ADDRESS 필드 ("local" 연결 유형 제외)	4
3.5. METHOD 필드	4
3.6. [OPTIONS] 필드 (선택 사항)	5
4. 이름에 특수 문자가 포함된 경우	5
5. 설정 파일 변경 사항 적용	5
참조	6

PostgreSQL 사용자 계정 접근설정

PostgreSQL의 클라이언트 인증 설정 파일인 `pg_hba.conf`는 PostgreSQL 서버 보안의 핵심적인 부분입니다. “HBA”는 “Host-Based Authentication”의 약자입니다.

1. 파일의 주요 역할

이 파일은 다음과 같은 중요한 사항들을 제어합니다:

- 어떤 호스트(클라이언트)가 서버에 연결할 수 있는지 (IP 주소, 호스트 이름 등)
- 클라이언트가 어떻게 인증되는지 (암호 방식, 운영체제 사용자 매핑 등)
- 클라이언트가 어떤 PostgreSQL 사용자 이름으로 접속할 수 있는지
- 클라이언트가 어떤 데이터베이스에 접근할 수 있는지

2. 레코드(규칙)의 형식

`pg_hba.conf` 파일의 각 레코드(규칙 한 줄)는 다음 중 하나의 형식으로 작성됩니다. 대문자로 표시된 부분은 실제 값으로 대체되어야 합니다.

```
local      DATABASE USER METHOD [OPTIONS]
host       DATABASE USER ADDRESS METHOD [OPTIONS]
hostssl    DATABASE USER ADDRESS METHOD [OPTIONS]
hostnossl  DATABASE USER ADDRESS METHOD [OPTIONS]
hostgssenc DATABASE USER ADDRESS METHOD [OPTIONS] (PostgreSQL 12부터 추가)
hostnogssenc DATABASE USER ADDRESS METHOD [OPTIONS] (PostgreSQL 12부터 추가)
```

3. 레코드의 필드 설명

3.1. 첫 번째 필드: 연결 유형 (Connection Type)

연결 유형은 클라이언트가 서버에 접속하는 방식을 지정합니다.

- `local`: 유닉스 도메인 소켓을 통한 연결입니다. 즉, 클라이언트와 서버가 같은 시스템에 있는 경우입니다. 이 경우 `ADDRESS` 필드는 사용되지 않습니다.
- `host`: 일반 TCP/IP 소켓 또는 SSL 암호화된 TCP/IP 소켓을 통한 연결입니다. 이 규칙은 SSL 연결과 비 SSL 연결 모두에 적용될 수 있습니다.
- `hostssl`: SSL 암호화된 TCP/IP 소켓을 통한 연결에만 적용됩니다. 이 규칙을 사용하려면 서버가 SSL을 지원하도록 설정되어 있어야 합니다 (`postgresql.conf` 파일의 `ssl = on`).
- `hostnossl`: SSL을 사용하지 않는 일반 TCP/IP 소켓을 통한 연결에만 적용됩니다.
- `hostgssenc`: GSSAPI로 암호화된 TCP/IP 소켓을 통한 연결입니다. GSSAPI는 Kerberos와 같은 외부 인증 시스템을 사용할 때 필요합니다. (PostgreSQL 12 이상)
- `hostnogssenc`: GSSAPI 암호화를 사용하지 않는 TCP/IP 소켓을 통한 연결입니다. (PostgreSQL 12 이상)

3.2. DATABASE 필드

이 규칙이 적용될 데이터베이스를 지정합니다.

- **all**: 모든 데이터베이스를 의미합니다. 단, **replication** 연결 요청과는 일치하지 않습니다. 복제 연결은 별도의 **replication** 키워드를 사용한 레코드로 명시적으로 허용해야 합니다.
- **sameuser**: 요청한 사용자와 이름이 같은 데이터베이스에만 적용됩니다.
- **samerole**: 요청한 사용자가 멤버인 역할(role)과 이름이 같은 데이터베이스에만 적용됩니다.
- **replication**: 물리적 복제 연결 요청에만 적용됩니다. (스트리밍 복제 또는 백업 시 사용)
- **데이터베이스_이름**: 특정 데이터베이스 이름을 명시합니다.
- **@파일이름**: 지정된 파일에 나열된 데이터베이스 이름들에 대해 적용됩니다. 파일 이름 앞에는 @ 기호를 붙여야 합니다. 파일 내에는 각 데이터베이스 이름이 한 줄씩 나열됩니다.
- **쉼표로 구분된 목록**: 여러 데이터베이스 이름이나 키워드를 쉼표로 구분하여 나열할 수 있습니다 (예: db1,db2,sameuser).

3.3. USER 필드

이 규칙이 적용될 PostgreSQL 사용자 이름을 지정합니다.

- **all**: 모든 사용자를 의미합니다.
- **사용자_이름**: 특정 사용자 이름을 명시합니다.
- **+그룹_이름**: 특정 그룹(역할)에 속한 모든 사용자를 의미합니다. 그룹 이름 앞에는 + 기호를 붙여야 합니다.
- **@파일이름**: 지정된 파일에 나열된 사용자 이름들에 대해 적용됩니다. 파일 이름 앞에는 @ 기호를 붙여야 합니다.
- **쉼표로 구분된 목록**: 여러 사용자 이름, 그룹 이름, 키워드를 쉼표로 구분하여 나열할 수 있습니다.

3.4. ADDRESS 필드 ("local" 연결 유형 제외)

이 규칙과 일치하는 클라이언트 호스트(들)를 지정합니다.

- **호스트 이름**: 예) myclient.example.com
- **.으로 시작하는 호스트 이름**: 호스트 이름의 접미사와 일치합니다. 예) .example.com은 foo.example.com과 bar.example.com 모두와 일치합니다.
- **IP 주소와 CIDR 마스크**: 예) 192.168.1.0/24 (IPv4), 2001:db8::/32 (IPv6). CIDR 마스크는 유효 비트 수를 나타냅니다 (IPv4: 0-32, IPv6: 0-128).
- **IP 주소와 넷마스크를 별도 컬럼으로**: 주석에서는 이 방법도 언급하지만, pg_hba.conf 파일의 표준 형식에서는 ADDRESS 단일 필드에 CIDR 표기법 (IP주소/마스크길이)을 사용하는 것이 일반적입니다. “별도 컬럼”이라는 표현은 개념적으로 IP 주소와 넷마스크 정보를 결합한다는 의미이거나, 특정 도구가 그런 입력을 받아 CIDR로 변환하여 파일에 기록하는 방식을 설명하는 것일 수 있습니다. 실제 파일에는 대부분 CIDR 형식으로 기재됩니다.
- **samehost**: 서버 자신의 모든 IP 주소와 일치합니다 (루프백 주소 포함, 서버의 다른 네트워크 인터페이스 IP 포함).
- **sameinet**: 서버가 직접 연결된 모든 서브넷의 주소와 일치합니다.

3.5. METHOD 필드

클라이언트 인증 방법을 지정합니다.

- **trust**: 무조건 연결을 허용합니다. 로컬 연결이나 매우 신뢰할 수 있는 네트워크에서만 주의해서 사용해야 합니다.
- **reject**: 무조건 연결을 거부합니다. 특정 클라이언트나 네트워크를 명시적으로 차단할 때 유용합니다.
- **md5**: 클라이언트가 MD5로 해시된 암호를 보내도록 요구합니다. 암호는 네트워크상에서 암호화되지 않은 형태로 전송되지 않습니다.
- **password**: 클라이언트가 일반 텍스트 암호를 보내도록 요구합니다. **이 방식은 암호가 네트워크 상에서 암호화되지 않고 전송되므로 보안에 매우 취약합니다. 테스트 환경이 아니라면 사용하지 않는 것이 좋습니다.**
- **scram-sha-256**: SCRAM-SHA-256 암호 인증을 사용합니다. md5보다 강력한 암호 기반 인증 방법이며, PostgreSQL 10부터 기본 암호화 방식으로 권장됩니다.
- **gss**: GSSAPI를 사용하여 인증합니다. Kerberos와 같은 외부 인증 시스템 연동에 사용됩니다. `hostgssenc` 또는 `hostnogssenc` 연결 유형과 함께 사용됩니다.
- **sspi**: SSPI를 사용하여 인증합니다. Windows 환경에서 사용되며, Active Directory 통합 등에 활용됩니다.
- **ident**: 클라이언트의 운영 체제 사용자 이름을 `ident` 서버를 통해 확인하여 인증합니다. TCP/IP 연결에서 사용되며, 보통 로컬호스트나 신뢰된 네트워크에서 사용됩니다. `pg_ident.conf` 파일에서 사용자 이름 매핑을 설정할 수 있습니다.
- **peer**: `local` (유닉스 도메인 소켓) 연결에서만 사용 가능하며, 클라이언트의 운영 체제 사용자 이름을 데이터베이스 사용자 이름으로 사용하여 인증합니다.
- **pam**: PAM(Pluggable Authentication Modules) 서비스를 사용하여 인증합니다. 리눅스/유닉스 시스템의 다양한 인증 메커니즘과 통합할 수 있습니다.
- **ldap**: LDAP 서버를 사용하여 인증합니다.
- **radius**: RADIUS 서버를 사용하여 인증합니다.
- **cert**: SSL 클라이언트 인증서를 사용하여 인증합니다. `hostssl` 연결 유형에서 사용됩니다.

주의: `password` 방식은 암호를 평문으로 전송하므로, `md5` 또는 `scram-sha-256` 사용이 강력히 권장됩니다.

3.6. [OPTIONS] 필드 (선택 사항)

`NAME=VALUE` 형식으로 인증 방법에 따른 추가 옵션을 설정합니다. 사용 가능한 옵션은 선택된 `METHOD`에 따라 다릅니다. 예를 들어, `ldap` 인증의 경우 LDAP 서버 주소, 바인딩 DN 등을 옵션으로 지정할 수 있습니다. 자세한 내용은 PostgreSQL 공식 문서의 “Client Authentication” 섹션을 참조해야 합니다.

4. 이름에 특수 문자가 포함된 경우

데이터베이스 이름이나 사용자 이름에 공백, 쉼표, 따옴표 또는 기타 특수 문자가 포함된 경우, 해당 이름을 큰따옴표(“”)로 묶어야 합니다. 키워드인 “`all`”, “`sameuser`”, “`samerole`”, “`replication`”을 큰따옴표로 묶으면 특수한 의미를 잃고 해당 문자열 자체를 이름으로 간주하게 됩니다. (예: “`all`”은 `all`이라는 이름의 데이터베이스나 사용자를 의미하게 됨)

5. 설정 파일 변경 사항 적용

`pg_hba.conf` 파일은 다음 경우에 다시 읽힙니다:

- PostgreSQL 서버가 시작될 때
- 서버 프로세스에 SIGHUP 신호를 보낼 때
- 명령어 `pg_ctl reload`를 실행할 때
- SQL 명령어 `SELECT pg_reload_conf();`를 실행할 때

따라서 파일을 편집한 후에는 위 방법 중 하나를 사용하여 서버가 변경 사항을 인식하도록 해야 합니다.

참조

- `pg_hba.conf` 파일의 규칙은 위에서부터 아래로 순서대로 평가되며, 가장 먼저 일치하는 규칙이 적용됩니다. 따라서 규칙의 순서가 매우 중요합니다.
- 호스트 이름을 사용하는 경우, DNS 조회에 따른 약간의 오버헤드가 발생할 수 있으며, DNS 서버의 신뢰성 및 보안에 의존하게 됩니다. 일반적으로 IP 주소를 사용하는 것이 더 빠르고 직접적입니다.

From: <https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link: https://atl.kr/dokuwiki/doku.php/postgresql_%EC%82%AC%EC%9A%A9%EC%9E%90_%EA%B3%84%EC%A0%95_%EC%A0%91%EA%B7%BC%EC%84%A4%EC%A0%95

Last update: 2025/05/22 01:28

