포스트그레스의 pg_hba.conf 활용

TPLUG(동명정보대학교 POWER LINUX USER GROUP) 주 효식 nogadax@kldp.org 2000년 4월4일

이 문서는 포스트그레스의 인증 방식등의 환경설정 파일인 pg_hba.conf를 설명하고, 이에 따른 사용상의 예제등을 제시하고자 한다. 또한, JDBC 를 이용한 connection 예제를 보여준다.

목 차

제 1 절 Domain Socket

pg_hba.conf 설정은 두가지의 방법이 있다. 첫째는 Unix Domain Socket 방법이고 둘째는 Internet Domain Socket 방법이다.

다음은 이들 방법의 설명이다.

socket 이란 unix 상의 프로세스간 통신을 위해 사용되는 메카니즘으로서, 이러한 통신에는 동일한 기계상의 프로세스간 통신과 네트워크 프로토콜을 이용하는 네트 워크상의 프로세스간 통신이 있다.

socket 은 domain 이라는 공통된 통신 특성에 따라 분류되며, 동일한 기계상의 프로세스간 통신을 Unix Domain Socket이라하고, 네트워크상의 프로세스간 통신 을 Internet Domain Socket (혹은 TCP/IP Domain Socket)라 한다.

Unix Domain Socket의 좋은 예로는 telnet 을 통한 telnet 디몬과 포스트그레스 디몬과의 통신이다.

Internet Domain Socket의 좋은 예로는 ODBC 나 JDBC 를 통해 원격지 즉, 다른 기계상에서 JDBC를 이용한 클라이언트 프로세스와 포스트그레스 디몬과의 통신이다. 그러므로 JDBC 혹은, ODBC 를 통해 포스트그레스 DB서버와의 connection 을 설정하 려할때 pg_hba.conf 의 설정은 Internet Domain Socket 방식으로 설정하여야 한다.

만약 127.0.0.1 인 IP Address 가 동일한 기계상의 프로세스간 통신을 위해 사용되었다 할지라도 127.0.0.1 은 네트워크상의 특수한 IP 이므로 이는 네트워크 프로토콜을 이용하는 네트워크상의 프로세스간 통신이다. 그러므로 이는 Internet Domain Socket 이 된다.

제 2 절 pg_hba.conf 설정

(postgresql Host-Based-Access Control)

보통 pg_hba.conf 는 $PGDATA 디렉토리에 존재한다. 예로서 포스트그레스의 디폴트 디렉토리는 /usr/local/pgsql이다. 그리고 PGDATA 의 값은 보통 /usr/local/pgsql/data 이다.

pg_hba.conf 파일은 일반적으로 레코드들로 이루어져 있으며 레코드란 특정한 형태를 지닌 하나의 라인이다. 라인의 제일 앞에 # 로 되어 있으면 이는 주석 처리가 되어 있는 것이다. 또한 라인이 공란이면 이는 무시되어 진다.

pg_hba.conf 의 레코드는 두가지로 분류되며 하나는 Unix Domain Socket 형이며 다른 하나는 Internet Domain Socket(TCP/IP Domain) 형이다.

2.1 Unix Domain Socket 설정

```
레코드 형식

 local  database  authentication method

 : local은 unix domain socket 방식의 연결을 허용한다.
   database 는 연결을 허용하려는 database 이며 모든
   database 에 연결을 허용하려면 all 을 적으면 된다.
   authentication method 는 사용자 인증방식을 설정한다.


예 1)

 local   all          trust

 : local 에서 모든 데이타베이스(all)에 대한 연결을 허용하며
   접속하려는 사용자들은 모두 믿을수 있는(trust) 사용자로
   인정하며 별다른 인증절차를 거치지 않고 바로 접속할 수 있다.


 psql을 통한 접속 :

  psql template1


예 2)

 local   template1    crypt

 : local 에서 template1 데이타베이스에만 접속을 허용하며 인증
   절차로 crypt 방식을 거친다. crypt는 사용자에 대한 패스워드를
   묻고 이 패스워드는 암호화되어 보내어진후 pg_shadow라는 파일에
   있는 패스워드와 비교한다. 비교된 패스워드가 맞아 떨어지면
   접속이 허용된다.
```

psql을 통한 접속 :

```
psql -u template1
```

(-u 옵션은 유저명과 패스워드를 묻는다)

2.2 Internet Domain Socket 설정

레코드 형식

```
host   databse  IP  IP_mask  auth  method
```

: jdbc 와 같은 커넥션을 통해 통신을 하는 client 들이 연결을 하려할 때 허용하거나 불허를 위해 설정한다. (즉, 다른 호스트에서 직접 포스트그레스의 사용자 계정으로 연결을 하려할 때)

```
host  template1  210.110.144.161  255.255.255.255  crypt
```

: 210.110.144.161 의 IP 만이 crypt 방식으로 template1 데이타베이스에 접속을 허용한다.

```
host  all        0.0.0.0          0.0.0.0          password
```

: 모든 데이타베이스와 모든 IP 주소들이 password 방식으로 접속을 할수가 있다.

2.3 Authentication method(인증 방법)

TRUST : 별다른 인증 절차없이 바로 허용한다.

REJECT : 무조건 접속을 거부한다.

CRYPT : 유저의 패스워드를 묻는다. 이때 패스워드는 암호화되어 포스트그레스에게 보내어진다.

PASSWORD : 유저의 패스워드를 묻는다. 이때 패스워드는 암호화되지 않은 문자 그대로 서버측인 포스트그레스에게 보내어진다.

다음의 인증 방법은 TCP/IP Domain 에만 해당된다.

krb4 : Kerberos V4
krb5 : Kerberos V5
ident : 클라이언트 상의 ident 서버가 사용되어지며 $PGDATA/pg_ident.conf 를 따로 설정하여야 한다.

또한, 네트워크 상의 커넥션을 하려할 때 논리적인 보안 터널을 사용하는 방법이 있다. 이는 ssh 보안 기법을 적용한 것으로 클라이언트와 포스트 그레스 서버 사이의 네트워크 컨넥션을 ssh를 이용하여 암화화할 수 있다. 적절한 적용으로 안전한 네트워크 컨넥션을 만들 수 있다.

```
참고사이트는 다음과 같다.

    http://www.heimhardt.de/htdocs/ssh.html
```

제 3 절 설정 테스트

3.1 포스트그레스 슈퍼유저 패스워드

테스트를 위해 postgres 의 슈퍼 유저 계정의 패스워드를 만들어야 한다. 이 패스워드는 단지 포스트그레스에 접속을 위한 것이며 telnet 에 로긴을 위한 패스워드는 아니다.

여기서 포스트그레스의 슈퍼유저 계정을 postgres 이고 패스워드는 nogadax라고 가정하며 이에 대한 절차는 다음과 같다.

```
단계 1)

[postgres@nogadax postgres]$ psql  template1
template1=> alter user postgres with password "nogadax";
...
template1=>\q

단계 2)

[postgres@nogadax postgres]$ vi $PGDATA/pg_hba.conf
...........
이부분에서 마지막 부분의 라인을 주석처리한 후
"local  all  crypt" 을  추가한다.
그런 후 저장하고 빠져 나온다.
...........

단계 3)

[postgres@nogadax postgres]$ psql -u template1
username : postgres
passwd   : nogadax
template1=>create user testusr with password "testusr" creatdb;
...........
template1=>select usename from pg_user;

usename
----------
 postgres
 testusr
 (2rows)

template1=>\q


단계 4)
```

```
[postgres@nogadax postgres]$ psql -u template1
  username : testusr
  password : testusr

 template1=>\q
[postgres@nogadax postgres]$
```

3.2 다양한 설정 예제

```
예 1) 다음은 가정 일반적인 설정이다.

  local  all  trust
  host   all  127.0.0.1  255.255.255.255  trust

local 를 통해 모든(all) 데이타베이스에 대한 접속을 허용하며
host 의 IP가 127.0.0.1 인 접속을 허용한다.

예 2) crypt 설정

crypt 에 대한 설정은 미리 유저명과 패스워드가 설정되어 있어야
한다. 또한 psql 접속시 -u 옵션을 적용하영야 한다.

  local  all  crypt
  host   all  210.110.144.161  255.255.255.255  crypt

예 3) reject 설정

  host  all  210.110.144.161  255.255.255.255  reject

210.110.144.161 의 IP 를 가진 곳으로부터의 접속을 허용하지
않는다. 즉, 거부한다(reject)

예 4) password 설정

 host  all  210.110.144.161 255.255.255.255  password
```

제 4 절 JDBC CONNECTION 테스트

1. 클라이언트는 power linux 2 를 사용하였고 서버로는 accel 리눅스6.2 를 사용하였다. 포스트그레스는 7.0 beta3 을 사용하였고 jdk1.2.2 를 사용하였다.

2. 테스트를 위해 클라이언트에 포스트그레스 JDBC 드라이버를 설치하여야 하며 설치시 주의할 점이 있다. 만약 jdk1.1.x 를 사용한다면 포스트그레스 JDBC1 을 사용하여야 하며 jdk1.2.x 를 사용한다면 포스트그레스 JDBC2 를 사용하여야 한다.

3. 클라이언트에 JDBC 드라이버 설치

먼저 포스트그레스 JDBC 드라이버를 클라이어트측의 적당한 디렉토리에 복사한다. 여기서는 /usr/local/lib/jdbc_post/ 디렉토리에 복사하였다. 그런 후 CLASSPATH 를 잡아준다. 다음은 CLASSPATH 설정 예이다.

```
CLASSPATH=/usr/local/lib/jdbc_post/postgresql.jar:.
export  CLASSPATH
```

4. 포스트그레스 서버의 pg_hba.conf 를 적절히 수정해주어야 한다. 테스트할 JDBC 소스 예제를 수행할 클라이언트의 IP 가 "210.110.144.161" 이라고 가정하고 포스트그레스 DB 서버의 IP 주소는 "210.110.144.162" 라고 가정한다. 테스트의 수행에 앞서 DB 서버측에 IP 주소가 "210.110.144.161" 인 클라이언트측에 대한 인증 설정을 하여야 한다. 설정은 다음과 같이 pg_hba.conf 에 다음 라인을 적절히 추가하여야 한다.

```
host all 210.110.144.161 255.255.255.255  crypt
```

5. 포스트그레스용의 적절한 예제 프로그램을 복사한 후 컴파일 한다. 여기서는 basic.java 를 가지고 테스트 한다. 포스트그레스 JDBC 관련 예제 프로그램은 포스트그레스의 소스 디렉토리의 /src/interfaces/jdbc/example 디렉토리에 있다.

또한 basic.java 소스의 제일 첫 라인의 "package example;" 을 적절히 처리한다. (주석처리나 삭제등..)

컴파일은 다음처럼 한다.

```
javac basic.java
```

실행은 다음처럼 한다.

java basic jdbc:postgresql://210.110.144.162/template1 postgres nogadax

위의 IP 어드레스 "210.110.144.162" 는 포스트그레스가 있는 서버의 IP 이고 "template1" 은 접속하여 테스트 하려는 데이타베이스명이다. "postgres" 는 접속을 하려하는 포스트그레스 의 DB 사용자명이고 "nogadax" 는 패스워드이다.

IP 는 자신의 서버에 맞게 적절히 변경해준다. 만약 하나의 서버상에서 테스트를 하려한다면 IP는 127.0.0.1 이며 pg_hba.conf 의 설정은 다음과 같다.

```
host all 127.0.0.1 255.255.255.255  crypt
```

제 5 절 기타

Domain Socket 에 대한 내용은 "Unix 의 내부 구조"(홍릉 과학 출판사,조유근 역)의 "프로세스간 통신" 부분을 참조하였다.

pg_hba.conf 는 포스트그레스의 Admin's Guide 의 "SECURITY" 부분을 참조하였다.

포스트그레스 JDBC 드라이버 관련 내용은 포스트그레스의 Programmer's Guide 의 "JDBC Interface" 부분을 참조하였다.