PostgreSQL + Apache + Zend 설치하기

정원영 suni00@kernel.pe.kr

v0.3, 2000년 2월 16일

DB로 PostgreSQL을 사용하며 Apache 웹서버에 PHP Zend를 모듈로 컴파일 하는 방법을 간략히 설명한다.

목 차

제 1 절 설치하기 전

- 설치환경은 RedHat 6.0 이상 버전으로 하겠다.
- 설치할 각각의 버전은 다음과 같다.
  - PostgreSQL : 6.5.3
  - Apache : 1.3.11
  - PHP : Zend beta 3
- 각각 원하는 버전으로 다음 사이트에서 구한다.
  - <http://www.postgresql.org>
  - <http://www.apache.org>
  - <http://www.php.net>

제 2 절 PostgreSQL 설치하기

2.1 postgres 계정 만들기

- postgres라는 계정과 그룹을 만든다. 기본적으로 RedHat 패키지는 useradd란 명령으로 그룹과 계정을 같이 만들어 준다.

```
# useradd postgres
# passwd postgres
```

2.2 백업 하기

- 처음 설치하는 경우는 **'2.3 컴파일및 설치'**로 넘어간다. 백업은 postgres 계정으로 작업한다.

```
$ pg_dumpall -z > db.out
```

2.3 컴파일및 설치

1. 컴파일 하기전 postmaster 데몬을 죽인다.

```
# ps -aux | grep postmaster
# kill PID (위에서 찾은 pid 번호)
```

2. PostgreSQL 소스를 /usr/local 방에서 푼다.

3. 풀면 postgresql-6.5.3이란 디렉토리가 생기는데 이름을 pgsql로 바꿔준다. (기존의 pgsql 디렉토리는 버전을 명시하며 이름을 바꿔준다.)

```
# mv postgresql-6.5.3 pgsql      (이름을 pgsql로 바꾼다)

# chown -R postgres.postgres pgsql (user와 group을 바꾼다)


# cd /usr/local/pgsql/src
# vi Makefile.custom             (Pentitum pro급 이상에서 최적화를
  CFLAG+= -m486                   위해 생성한다)

# ./configure --with-mb=EUC_KR    (한글 사용을 위해)

# cd /usr/local/pgsql/doc
# make install                   (맨 페이지, HTML 문서파일 설치)

# cd /usr/local/pgsql/src
# make all                       (컴파일)
```

4. 'All of PostgreSQL is successfully made. Ready to install' 란 메세지가 보이면 성공.

```
# su postgres                    (postgres 계정으로 작업하기위해)
$ make install                   (설치)
```

2.4 설치후 작업

1. 라이브러리 추가

```
# vi /etc/ld.so.conf
  /usr/local/pgsql/lib
# /sbin/ldconfig
```

2. 환경변수 추가

각 개인의 환경변수를 설정하기위해 개인의 .bash_profile에 다음을 추가한다.
(DB를 사용할 계정의 환경변수를 설정 - postgres 계정에도 적용시킨다.)

```
PATH=$PATH:/usr/local/pgsql/bin
MANPATH=$MANPATH:/usr/local/pgsql/man
PGLIB=/usr/local/pgsql/lib
PGDATA=/usr/local/pgsql/data
export PATH MANPATH PGLIB PGDATA
```

3. 환경 변수 적용

각 계정에서 실행해서 환경변수를 적용시킨다.

```
$ source .bash_profile
```

4. 데이터베이스 초기화 하기

postgres계정으로 작업해야한다.

```
$ initdb -e EUC_KR
```

2.5 postmaster 데몬 뛰우기

- postgres 계정으로 뛰워야 한다.

```
$ postmaster -i -S -D /usr/local/pgsql/data
```

2.6 백업 데이터 복구

- 백업된 파일 db.out을 복구하기 위해 다음과 같이 postgres 계정으로 작업한다.

```
$ psql -e template1 < db.out
```

제 3 절 Apache + PHP 설치하기

1. 환경설정및 컴파일

아파치와 PHP모두 /usr/local 디렉토리에 풀었으며 아파치의 소스 디렉토리는 /usr/local/apache_1.3.11이며 PHP의 소스 디렉토리는 /usr/local/php-4.0b3이다.
모든건 root 계정으로 작업한다.

```
# cd /usr/local/apache_1.3.11     (아파치 소스 디렉토리로 이동)

# ./configure --prefix=/etc/httpd (컴파일후 생성 파일의 경로는
                                    /etc/httpd 이다)


# cd /usr/local/php-4.0b3         (php 소스 디렉토리로 이동)

# ./configure --with-apache=/usr/local/apache_1.3.11
              --with-pgsql=/usr/local/pgsql
              --enable-track-vars
            (아파치와 postgres의 경로인데 한줄에 이어 써야한다)
# make
# make install                    (php 설치)

# cp php.ini-dist /usr/local/lib/php.ini (php 옵션들)


# cd /usr/local/apache_1.3.11     (아파치 디렉토리로 이동)
# ./configure --prefix=/etc/httpd
              --activate-module=src/modules/php4/libphp4.a
        (php를 모듈로 포함시킴, 역시 한줄로 이어 써야한다)
# make
# make install                    (아파치 설치)
```

2. 설치 후

```
설치후 아파치 환경설정 파일인 /etc/httpd/conf/httpd.conf 파일을 열어
다음과 같이 되어있는지 확인하고 주석을 풀어준다.

        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
```

제 4 절 설치 테스트 하기

지금부터 쿠키를 이용한 PHP + PostgreSQL용 간단한 카운터를 만들어보며 설치가 제대로 되었는지 확인해 보자. 사용자 계정은 suni00으로 가정하겠다.

4.1 일반 사용자에게 DB 권한주기

- postgres 계정으로 일반 사용자에게 DB 권한을 주어야지 DB를 생성할 수 있다.

```
$ su - postgres                   (사용자를 postgres로 바꾼다)
$ createuser suni00               (권한을 주고픈 계정을 적는다)
```

DB 생성 권한은 꼭 주어야 한다.

4.2 DB 생성및 테이블 만들기

- postgres가 준 권한으로 일반사용자가 다음과 같은 DB및 테이블을 생성한다.

```
$ su - suni00                         (일반계정 사용자로 바꾼다)
$ createdb test                       (test란 DB를 만든다)
$ psql test                           (test DB에 연결)

test=> create table counter ( count int );  (table 생성)
test=> insert into counter values ( 0 );    (0을 입력)
test=> \q                                   (DB 빠져나감)
```

4.3 PHP 프로그래밍 하기

- 다음과 같은 내용의 count.php 파일을 만든다.

```
<?
        $connect=pg_connect("dbname=test user=suni00");
        $result=pg_exec($connect,"select * from counter");

        $count_num = pg_result($result,0,"count");

        if (!$counted) {
                $count_num++;
                pg_exec($connect,"update counter set count=$count_num");
        }

        echo $count_num;
?>
```

이제 웹에서 이 파일을 읽어보면 reload 할 때마다 하나씩 카운트가 증가하는걸 알 수 있다.

4.4 쿠키 사용하기

- 앞에서 만든 카운터는 reload 할때마다 증가하므로 그리 정확하지 않은 단점이 있다. 그래서 다음과 같이 쿠키를 이용하여 약간이나마 신뢰할 수 있게 해보자. 쿠키는 모든 태그보다 위에있어야 하기 때문에 < html > 시작 태그보다 윗줄에 다음과 같이 넣어준다.

```
<? if (!$counted) { setcookie("counted",1,time()+3600); } ?>
```

위 내용을 대충 설명하면 3600초(1시간)동안 $counted 값을 1로 만들어 준다. 그러므로 한번 접속한 후 1시간 뒤에 접속해야지만 카운터는 증가한다.