

Linux Server (for Centos 7.x)

- APM -

Copyright @ 2016 MajunSoft co.,Ltd

소 속	IDC실
이 름	신용우 매니저
E-mail	tech@tongkni.co.kr

통큰아이

INDEX

- 1. 개요 3
- 2. APM 소스 컴파일 설치 스크립트를 이용하여 설치하기. 4
- 3. 웹 서버 구축하기. 15
 - 3.1 httpd.conf 환경 설정. 15
 - 3.2 가상 호스트 설정. 18
- 4. 활용하기. 19
 - 4.1 phpMyAdmin 설치하기.. 19
 - 4.2 Log Rotate 설정하기. 29
 - 4.3 SSL 서버 설치하기. 30
- 5. 다른 웹프로그래밍(nginx)을 이용하여 서버 구축하기. 35
 - 3.1 nginx 설치. 35
 - 3.2 nginx 설정. 41

1. 개요.

- APM이란 Apache(웹 서버)+Php(웹 프로그래밍 언어 프로그램)+Mysql(데이터베이스 서버)의 앞글자를 따서 만들어진 약어로, 웹 서비스를 용도로 서버를 운영하기 위해 설치하는 서비스입니다.
- Apache는 리눅스에서 사용하는 가장 안정적이고 유명한 소프트웨어입니다. 보통 PHP언어와 Mysql DB와 연동하여 운영하며, 각각 제작한 곳이 다르기 때문에 소프트웨어가 잘 연동되도록 설치해야 합니다.
- APM 설치 방법에는 RPM 설치, 소스 컴파일 설치 방식이 있습니다. 소스 컴파일 방식은 RPM 설치 방식에 비해 설정파일이나 실행파일 위치 지정 및 설치 옵션을 부여하기가 용이합니다. 따라서 본 매뉴얼에서는 소스 컴파일 방식으로 진행하며, 통큰아이에서 자체 제작한 자동 설치 스크립트(CentOS 6.2에 최적화)를 사용합니다.
- 아래는 APM을 이용한 웹 서버 구축 절차입니다.
 - 1 APM 소스 설치 스크립트를 이용하여 설치.
 - 2 설치 및 연동이 정상적으로 되었는지 확인.
 - 3 Apache 설정.
- Apache의 주 설정파일인 httpd.conf 파일의 내용은 크게 3개의 section으로 분류됩니다.
 - 1 **Global Environment** : 아파치 웹 서버의 전반적인 환경 설정 파트입니다. 서버 구동 방식, 아파치 서버의 홈 디렉토리 경로, 웹 서버에 접근 가능한 클라이언트 허용 개수 등의 데몬 관련 설정을 담당합니다.
 - 2 **Main Server configuration** : 아파치 웹 서버의 메인 웹 사이트가 사용할 값들을 설정하는 파트입니다. 웹 서버의 기본 포트, 메인 웹 사이트의 웹 문서 경로, 기본 페이지 파일 지정 및 순서 등의 설정을 담당합니다.
 - 3 **Virtual Hosts** : 한 시스템에서 여러 개의 웹 사이트를 운영하고자 할 때 설정하는 파트입니다. 각 가상 호스트별로 홈 디렉토리, 도메인 이름, 로그파일 경로 등을 지정할 수 있습니다.
- 본 매뉴얼은 WEB 서비스를 하기 위한 APM 설치 및 구축 방법과 추가적으로 Apache와 비슷한 다른 웹 서버 소프트웨어인 nginx설치 방법과 사용법을 소개하였습니다.

2. APM 소스 컴파일 설치 스크립트를 사용하여 설치하기.

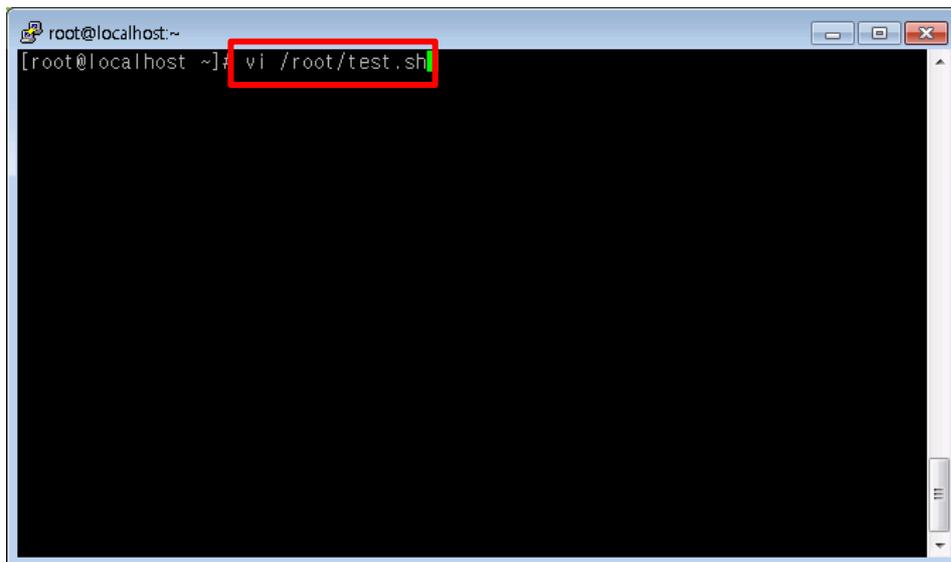
- 1 지금부터 통큰아이에서 자체 제작한 자동 설치 스크립트를 이용하여 APM을 설치합니다. 아래는 자동 설치 스크립트로 설치되는 소프트웨어 목록입니다.

※ OS 설치 시 기본적인 구성요소가 설치되어 있지 않거나 각 스크립트에 맞지 않는 버전의 OS일 경우 설치가 실패할 수 있습니다.

- yum update - 커널, 라이브러리 및 패키지 업데이트.
- 서버 시간을 time.bora.ne(ntp 서버)와 동기화
- Apache 2.4.20 버전 설치.
- Mysql 5.6.10 버전 설치. (언어셋=utf-8)
- PHP 5.6.18 버전 설치.
- ZendGuardLoader(for php 5.6)설치.
- 방화벽 (iptables)에 21번, 25번, 443번, 80번, 3306번, 53번 포트 허용정책 추가.
- MOD_URL 설치.

- 2 root 홈 디렉토리에 APM 설치 스크립트 파일을 생성합니다.

- vi /root/test.sh



- 3 편집기에서 “i”를 눌러 입력모드로 전환하고 통큰아이에서 자체 제작한 아래 스크립트를 붙여넣은 후 “:wq!”를 입력하여 저장합니다. (붙여넣기 : 입력모드 전환 후 Shift + Insert)

※ 본 매뉴얼은 CentOS 7.x 커널 기반으로 작성이 되었으므로, 이전 버전 사용자분께서는 Centos 6.X 5.X버전의 매뉴얼을 참고하셔서 진행하시기 바랍니다.

CentOS 7.x 버전

```
#!/bin/bash
yum -y install wget make gcc*
yum -y update

clear
echo "#####"
echo "##### 설치 시작 합니다. #####"
echo "##### 설치를 시작 합니다. #####" > /root/autoscripterr.log

#중요패키지 설치
yum -y install autoconf libjpeg libjpeg-devel libpng libpng-devel libtiff-devel libtiff freetype freetype-devel libxml-devel libxml libxml2 libxml2-
devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel bzip2-libs ncurses ncurses-devel curl-devel e2fsprogs e2fsprogs-
devel krb5 krb5-devel libidn libidn-devel openssl* libtool libtool-libs libtool-ltdl libtool-ltdl-devel openldap openldap-devel nss_ldap
openldap-clients openldap-servers bison gd gd2 gd-devel* gd2-devel gdbm-devel gdbm pcre pcre-devel perl libaio libc-client libc-client-
devel mhash mhash-devel libmcrypt libmcrypt-devel imap-devel imap uw-imap-devel expat-devel expat gmp* lemon libssh2 libssh2-devel
#iptables 설치
yum -y install system-config-firewall
echo "#####중요 패키지 설치 #####"
echo "#####중요 패키지 설치 #####" >> /root/autoscripterr.log

# 시간 동기화
/usr/bin/rdate -s time.bora.net && /sbin/clock -w 2>> /root/autoscripterr.log

echo "##### MySQL 설치 Start. #####"
echo "##### MySQL 설치 Start . #####" >> /root/autoscripterr.log

yum -y remove mysql*
yum -y install cmake.x86_64
yum -y install ncurses-devel.x86_64

# mysql 설치
cd /usr/local/src/
wget -O mysql-5.6.10.tar.gz http://www.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.10.tar.gz/from/http://cdn.mysql.com/
tar xvf mysql-5.6.10.tar.gz
cd /usr/local/src/mysql-5.6.10

echo "##### MySQL 컴파일 . #####"
echo "##### MySQL 컴파일 . #####" >> /root/autoscripterr.log

cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \
-DMYSQL_DATADIR=/usr/local/mysql/data \
-DSYSCONFDIR=/etc/mysql \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DENABLED_LOCAL_INFILE=1 \
-DWITH_EXTRA_CHARSETS=all

## MySQL 설치
make && make install

\cp -rf /usr/local/mysql/support-files/my-default.cnf /etc/my.cnf

echo "basedir = /usr/local/mysql" >> /etc/my.cnf
echo "datadir = /usr/local/mysql/data" >> /etc/my.cnf
echo "port = 3306" >> /etc/my.cnf
echo "server_id = 1" >> /etc/my.cnf
echo "socket = /tmp/mysql.sock" >> /etc/my.cnf
echo "explicit_defaults_for_timestamp=true" >> /etc/my.cnf

groupadd mysql
useradd -r -g mysql mysql

cd /usr/local
chown -R mysql:mysql mysql
cd mysql
scripts/mysql_install_db \
-user=mysql \
```

```

-datadir=/usr/local/mysql/data \
-basedir=/usr/local/mysql

echo "#####      MySQL 설치 완료 .      #####"
echo "#####      MySQL 설치 완료 .      #####" >> /root/autoscripterr.log

cp support-files/mysql.server /etc/init.d/mysqld
chmod 755 /etc/init.d/mysqld
/etc/init.d/mysqld start

systemctl enable mysqld.service

## MySQL Path 등록
cd
echo "export PATH=$PATH:$HOME/bin:/usr/local/mysql/bin" >> /etc/profile
cat .bash_profile | sed 's\PATH=$PATH:$HOME/bin\PATH=$PATH:$HOME/bin:/usr/local/mysql/bin \g' > .bash_profile.1
mv -f .bash_profile.1 .bash_profile
source .bash_profile

echo "#####      아파치 설치 Start .      #####"
echo "#####      아파치 설치 Start .      #####" >> /root/autoscripterr.log

#아파치 다운

cd /usr/local/src/
wget http://mirror.apache-kr.org//httpd/httpd-2.4.18.tar.bz2
tar -jxf httpd-2.4.18.tar.bz2

#pcre 설치
cd /usr/local/src/
wget -O pcre-8.34.tar.gz https://sourceforge.net/projects/pcre/files/pcre/8.34/pcre-8.34.tar.gz/download
tar zxvf pcre-8.34.tar.gz
cd pcre-8.34/
./configure --libdir=/usr/local/lib64
make && make install

ln -s /usr/local/lib64/libpcre.so.1 /lib

## apr 1.5.3 설치
cd /usr/local/src/
wget https://archive.apache.org/dist/apr/apr-1.5.1.tar.bz2
tar -jxf apr-1.5.1.tar.bz2
cd /usr/local/src/apr-1.5.1
./configure --prefix=/usr/local/apr
make
make install

## apr-iconv 설치
cd /usr/local/src/
wget https://archive.apache.org/dist/apr/apr-iconv-1.2.1.tar.gz
tar zxvf apr-iconv-1.2.1.tar.gz
cd /usr/local/src/apr-iconv-1.2.1
./configure --prefix=/usr/local/apr-iconv --with-apr=/usr/local/apr
make
make install

## apr-util 설치
cd /usr/local/src/
wget https://archive.apache.org/dist/apr/apr-util-1.5.3.tar.bz2
tar -jxf apr-util-1.5.3.tar.bz2
cd /usr/local/src/apr-util-1.5.3
./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr --with-apr-iconv=/usr/local/apr-iconv/bin/apriconv
make
make install

echo "#####      아파치 컴파일 .      #####"
echo "#####      아파치 컴파일 .      #####" >> /root/autoscripterr.log

cd /usr/local/src/httpd-2.4.18/

groupadd apache
useradd -g apache apache

```

```

./configure --prefix=/usr/local/apache \
--enable-authn-alias \
--enable-cache \
--enable-file-cache \
--enable-mem-cache \
--enable-disk-cache \
--enable-mods-shared=most \
--enable-proxy=shared \
--enable-proxy-ajp=shared \
--enable-proxy-ajp=shared \
--enable-dav=shared \
--with-apr=/usr/local/apr/ \
--with-apr-util=/usr/local/apr-util/ \
--enable-ssl \
--enable-dav \
--enable-cgi \
--with-mpm=prefork \
--enable-cgid \
--enable-vhost-alias \
--enable-rewrite \
--enable-forward=shared \
--enable-proxy-connect=shared \
--enable-proxy-http=shared \
--enable-deflate=shared \
--enable-so \
--enable-headers=shared \
--enable-include=shared \
--enable-rewrite=shared

## 아파치 설치
make&&make install

\cp /usr/local/apache/bin/apachectl /etc/init.d/httpd

cat /etc/init.d/httpd | sed 's@#!/bin/sh@#!/bin/sh\n\n# chkconfig: 2345 90\n# description: init file for apachectl server\n# processname: /usr/local/apache/bin/apachectl\n# config: /usr/local/apache/conf/httpd.conf\n# pidfile: /usr/local/apache/logs/httpd.pid@g' > /etc/init.d/httpd_t

mv -f /etc/init.d/httpd_t /etc/init.d/httpd
chmod 755 /etc/init.d/httpd

chkconfig --add httpd
chkconfig --level 3 httpd on

systemctl enable httpd
systemctl enable httpd.service

#아파치 동작을 위한 기본세팅
cat /usr/local/apache/conf/httpd.conf | sed 's\#ServerName www.example.com:80\ServerName 127.0.1.1\g' > /usr/local/apache/conf/httpd_t.conf
mv -f /usr/local/apache/conf/httpd_t.conf /usr/local/apache/conf/httpd.conf

cat /usr/local/apache/conf/httpd.conf | sed 's\Deny from all\allow from all\g' > /usr/local/apache/conf/httpd_t.conf
mv -f /usr/local/apache/conf/httpd_t.conf /usr/local/apache/conf/httpd.conf

#iptables 설정
service iptables stop

cat /etc/sysconfig/iptables | sed 's\A INPUT j REJECT --reject-with icmp-host-prohibited\A INPUT -m state --state NEW -m tcp -p tcp --dport 21 j ACCEPT\g' > /etc/sysconfig/iptables_t
mv -f /etc/sysconfig/iptables_t /etc/sysconfig/iptables 2>> /root/autoscripterr.log

cat /etc/sysconfig/iptables | sed 's\COMMIT\A INPUT -m state --state NEW -m tcp -p tcp --dport 20 j ACCEPT\g' > /etc/sysconfig/iptables_t
mv -f /etc/sysconfig/iptables_t /etc/sysconfig/iptables 2>> /root/autoscripterr.log

echo "-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT -m state --state NEW -m tcp -p tcp --dport 25 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT -m state --state NEW -m udp -p udp --dport 53 j ACCEPT" >> /etc/sysconfig/iptables
echo "-A INPUT j REJECT --reject-with icmp-host-prohibited" >> /etc/sysconfig/iptables

```

```

echo "COMMIT" >> /etc/sysconfig/iptables

echo "##### iptables 수정 완료. #####"
echo "##### iptables 수정 완료. #####" >> /root/autoscripterr.log

service iptables restart 2>> /root/autoscripterr.log
systemctl enable iptables.service

echo "##### 아파치 설치 완료. #####"
echo "##### 아파치 설치 완료. #####" >> /root/autoscripterr.log

/etc/rc.d/init.d/httpd restart

## PHP 설치
echo "##### PHP 설치 Start . #####"
echo "##### PHP 설치 Start . #####" >> /root/autoscripterr.log

cd /usr/local/src
wget -O php-5.6.18.tar.gz http://kr1.php.net/get/php-5.6.18.tar.gz/from/this/mirror
tar xzvf php-5.6.18.tar.gz

#libmcrypt 설치
cd /usr/local/src
wget -O libmcrypt-2.5.8.tar.gz https://sourceforge.net/projects/mcrypt/files/Libmcrypt/2.5.8/libmcrypt-2.5.8.tar.gz/download
tar xzvf libmcrypt-2.5.8.tar.gz
cp -a libmcrypt-2.5.8 /usr/local
cd /usr/local/libmcrypt-2.5.8
./configure --prefix=/usr/local/
make
make check
make install

echo "##### libmcrypt 설치 완료 . #####"
echo "##### libmcrypt 설치 완료 . #####" >> /root/autoscripterr.log

#curl 설치
cd /usr/local/src
wget https://curl.haxx.se/download/curl-7.48.0.tar.gz
tar xzvf curl-7.48.0.tar.gz
cd curl-7.48.0
./configure --prefix=/usr/local/curl --with-ssl=/usr/local/ssl --with-libssh2=/usr/include/
make
make install

echo "##### curl 설치 완료 . #####"
echo "##### curl 설치 완료 . #####" >> /root/autoscripterr.log

#re2c 설치
cd /usr/local/src
wget -O re2c-0.13.5.tar.gz https://sourceforge.net/projects/re2c/files/old/re2c-0.13.5.tar.gz/download
tar xzvf re2c-0.13.5.tar.gz 2>> /root/autoscripterr.log
cd re2c-0.13.5
./configure
make && make install

echo "##### re2c 설치 완료 . #####"
echo "##### re2c 설치 완료 . #####" >> /root/autoscripterr.log

#flex 설치
cd /usr/local/src
wget http://dfn.dl.sourceforge.net/sourceforge/flex/flex-2.5.4a.tar.gz
tar xzvf flex-2.5.4a.tar.gz
cd /usr/local/src/flex-2.5.4
./configure --prefix=/usr/local
make && make install

echo "##### flex 설치 완료 . #####"
echo "##### flex 설치 완료 . #####" >> /root/autoscripterr.log

cd /usr/local/src/php-5.6.18

./configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache/bin/apxs --with-config-file-path=/usr/local/apache/conf --with-mysql=/usr/local/mysql --with-curl=/usr/local/curl --enable-calendar --enable-shmop --enable-ftp --enable-sockets --disable-cgi --with-gd

```

```

-with-png-dir=/usr/lib --with-zlib-dir --with-jpeg-dir=/usr/lib --with-freetype-dir=/usr/lib --with-iconv --enable-mbstring --with-mcrypt=/usr/local
-with-openssl --with-mysql=/usr/local/mysql/bin/mysql_config

make && 2>> /root/autoscripterr.log
make install && 2>> /root/autoscripterr.log

echo "#####      PHP make 및 make install 완료.      #####"
echo "#####      PHP make 및 make install 완료.      #####" >> /root/autoscripterr.log
cp php.ini-production /usr/local/apache/conf/php.ini 2>> /root/autoscripterr.log

#php 기본환경설정
echo "<?php" > /usr/local/apache/htdocs/index.php
echo "echo phpinfo();" >> /usr/local/apache/htdocs/index.php
echo ">" >> /usr/local/apache/htdocs/index.php

cat /usr/local/apache/conf/php.ini | sed 's/register_globals = Off/register_globals = On/g' > /usr/local/apache/conf/php_t.ini
mv -f /usr/local/apache/conf/php_t.ini /usr/local/apache/conf/php.ini 2>> /root/autoscripterr.log

cat /usr/local/apache/conf/php.ini | sed 's/magic_quotes_gpc = Off/magic_quotes_gpc = On/g' > /usr/local/apache/conf/php_t.ini
mv -f /usr/local/apache/conf/php_t.ini /usr/local/apache/conf/php.ini 2>> /root/autoscripterr.log

cat /usr/local/apache/conf/httpd.conf | sed 's/#AddEncoding x-compress .Z/AddType application/x-httpd-php .php .htm .html/g' >
/usr/local/apache/conf/httpd_t.conf
mv -f /usr/local/apache/conf/httpd_t.conf /usr/local/apache/conf/httpd.conf 2>> /root/autoscripterr.log

cat /usr/local/apache/conf/httpd.conf | sed 's/#AddEncoding x-gzip .gz .tgz/AddType application/x-httpd-php-source .phps/g' >
/usr/local/apache/conf/httpd_t.conf
mv -f /usr/local/apache/conf/httpd_t.conf /usr/local/apache/conf/httpd.conf 2>> /root/autoscripterr.log

service httpd restart

setenforce 0
systemctl disable firewalld.service

#zendGuard 설치
echo "#####      ZendGuard 설치      #####"
echo "#####      ZendGuard 설치      #####" >> /root/autoscripterr.log

cd /usr/local/src
wget http://downloads.zend.com/guard/7.0.0/zend-loader-php5.6-linux-x86_64.tar.gz
tar xzvf zend-loader-php5.6-linux-x86_64.tar.gz
cd /usr/local/src/zend-loader-php5.6-linux-x86_64
cp ZendGuardLoader.so /usr/local/php/lib/

cat >> /usr/local/apache/conf/php.ini << DOCUMENT
[Zend]
zend_extension=/usr/local/php/lib/ZendGuardLoader.so
DOCUMENT

service httpd restart

## mod_url 설치
cd /usr/local/src
wget http://caos.kldp.net/frs/download.php/6015/mod_url-apache-1.6.2.6.tar.bz2
bzip2 -d mod_url-apache-1.6.2.6.tar.bz2
tar -xvf mod_url-apache-1.6.2.6.tar

cd mod_url-apache
/usr/local/apache/bin/apxs -i -a -c mod_url.c

cat >> /usr/local/apache/conf/httpd.conf << DOCUMENT
<IfModule mod_url.c>
    CheckURL On
</IfModule>
DOCUMENT

service httpd restart

#방화벽 재시작과 등록
/bin/systemctl start iptables.service
systemctl enable iptables.service

echo "#####      모든 설치 완료. 수고하셨습니다.      #####"
echo "#####      모든 설치 완료. 수고하셨습니다.      #####" >> /root/autoscripterr.log

```

- 4 설치 스크립트를 실행할 수 있도록 권한을 부여합니다.
 → `chmod 700 /root/test.sh`

```

root@localhost:~# vi /root/test.sh
root@localhost:~# chmod 700 /root/test.sh
root@localhost:~# ls -al /root/
합계 100
dr-xr-x---. 2 root root 4096 2012-02-20 17:18 .
dr-xr-xr-x. 24 root root 4096 2012-02-20 14:13 ..
-rw-----. 1 root root 1297 2012-02-17 16:09 .bash_history
-rw-r--r--. 1 root root 18 2009-05-20 19:45 .bash_logout
-rw-r--r--. 1 root root 176 2009-05-20 19:45 .bash_profile
-rw-r--r--. 1 root root 176 2004-09-23 12:59 .bashrc
-rw-r--r--. 1 root root 100 2004-09-23 12:59 .cshrc
-rw-r--r--. 1 root root 129 2004-12-04 06:42 .tcshrc
-rw-----. 1 root root 5755 2012-02-20 17:04 .viminfo
-rw-----. 1 root root 1279 2012-02-16 19:19 anaconda-ks.cfg
-rw-r--r--. 1 root root 28560 2012-02-16 19:19 install.log
-rw-r--r--. 1 root root 7352 2012-02-16 19:17 install_log_syslog
-rwx-----. 1 root root 13814 2012-02-20 17:18 test.sh
root@localhost:~#
  
```

- 5 설치 스크립트를 실행합니다. 자동으로 update 및 APM이 설치되며, 설치 환경에 따라 1~3시간 정도 소요됩니다.
 → `/root/test.sh`

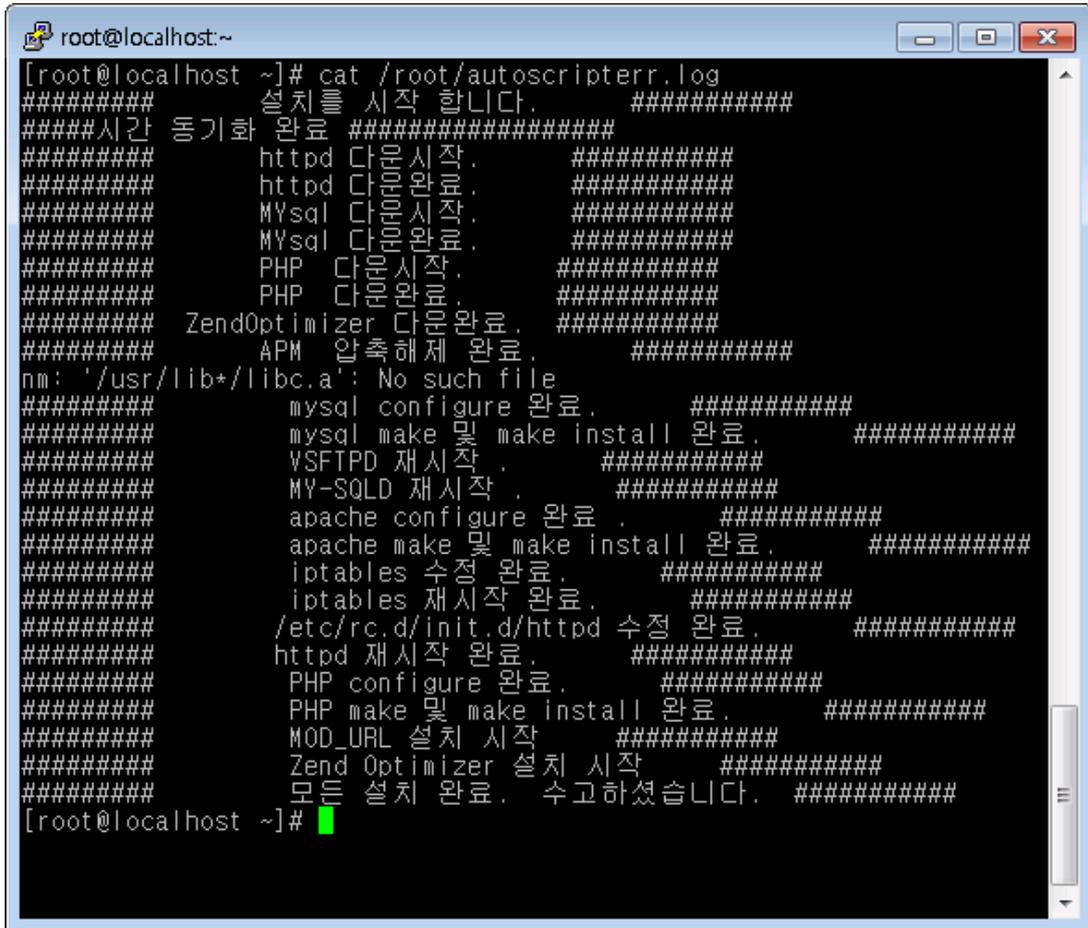
```

selinux-policy          noarch  3.7.19-126.el6_2.6      updates  773 k
selinux-policy-targeted noarch  3.7.19-126.el6_2.6      updates  2.5 M
strace                  i686   4.5.19-1.11.el6_2.1     updates  167 k

Transaction Summary
-----
Install      2 Package(s)
Upgrade     40 Package(s)

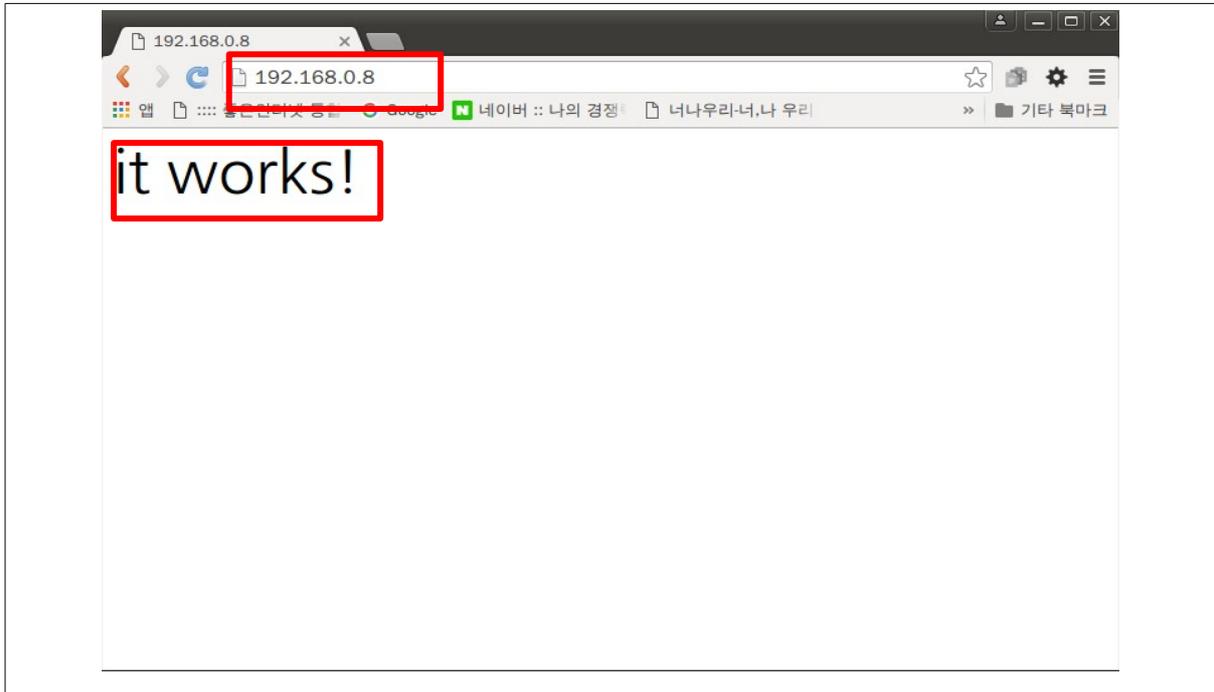
Total download size: 114 M
Downloading Packages:
(1/42): at-3.1.10-43.el6_2.1.i686.rpm | 59 kB  00:00
(2/42): bind-libs-9.7.3-8.P3.el6_2.2.i686.rpm | 850 kB  00:00
(3/42): bind-utils-9.7.3-8.P3.el6_2.2.i686.rpm | 177 kB  00:00
(4/42): dhclient-4.1.1-25.P1.el6_2.1.i686.rpm | 314 kB  00:00
(5/42): dhcp-common-4.1.1-25.P1.el6_2.1.i686.rpm | 139 kB  00:00
(6/42): freetype-2.3.11-6.el6_1.8.i686.rpm | 362 kB  00:00
(7/42): freetype-devel-2.3.11-6.el6_1.8.i686.rpm | 364 kB  00:00
(8/42): glibc-2.12-1.47.el6_2.5.i686.rpm | 4.3 MB  00:00
(9/42): glibc-common-2.12-1.47.el6_2.5.i686.rpm | 14 MB  00:01
(10/42): glibc-devel-2.12-1.47.el6_2.5.i686.rpm | 967 kB  00:00
(11/42): glibc-headers-2.12-1.47.el6_2.5.i686.rpm | 605 kB  00:00
(12/42): jasper-libs-1.900.1-15.el6_1.1.i686.rpm | 136 kB  00:00
(13/42): java-1.6.0-op (35%) 76% [===== ] 6.8 MB/s | 20 MB  00:00 ETA
  
```

- 6 APM 설치 로그파일을 확인하여 설치가 정상적으로 완료되었는지 확인합니다.
→ cat /root/autoscripterr.log

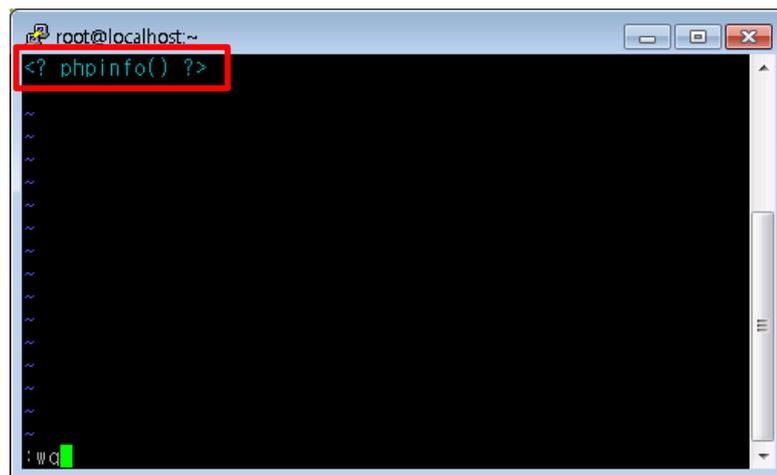


```
root@localhost:~  
[root@localhost ~]# cat /root/autoscripterr.log  
#####  
##### 설치를 시작 합니다. #####  
#####시간 동기화 완료 #####  
##### httpd 다운 시작. #####  
##### httpd 다운 완료. #####  
##### Mysql 다운 시작. #####  
##### Mysql 다운 완료. #####  
##### PHP 다운 시작. #####  
##### PHP 다운 완료. #####  
##### ZendOptimizer 다운 완료. #####  
##### APM 압축해제 완료. #####  
nm: '/usr/lib*/libc.a': No such file  
##### mysql configure 완료. #####  
##### mysql make 및 make install 완료. #####  
##### VSFTPD 재시작. #####  
##### MY-SQLD 재시작. #####  
##### apache configure 완료. #####  
##### apache make 및 make install 완료. #####  
##### iptables 수정 완료. #####  
##### iptables 재시작 완료. #####  
##### /etc/rc.d/init.d/httpd 수정 완료. #####  
##### httpd 재시작 완료. #####  
##### PHP configure 완료. #####  
##### PHP make 및 make install 완료. #####  
##### MOD_URL 설치 시작 #####  
##### Zend Optimizer 설치 시작 #####  
##### 모든 설치 완료. 수고하셨습니다. #####  
[root@localhost ~]#
```

- 7 인터넷 브라우저 창에 웹 서버 IP를 입력하면 "It works"라고 출력되는지 확인하여 Apache가 정상적으로 구동되었는지 확인합니다.



- 8 phinfo 기능을 이용하여 다른 소프트웨어가 정상적으로 설치 및 연동되었는지 확인하기 위해 현재 웹사이트 기본페이지인 /usr/local/apache/htdocs/index.html을 편집합니다.
- ➔ vi /usr/local/apache/htdocs/index.html
 - ➔ “i”를 눌러 편집 모드로 진입 후 내용을 모두 지우고 아래 내용 입력.
<? phinfo() ?>
 - ➔ <ESC>를 누른 후 “:wq!”를 입력하여 저장 후 종료.



- 9 다시 인터넷 브라우저 창에 서버 IP를 입력하면 이전과는 달리 아래 화면처럼 php 프로그램의 환경 설정에 관한 모든 정보를 테이블 형태로 출력됩니다. 여기서 각 소프트웨어가 정상적으로 설치 및 연

동되었는지 대략적으로 확인할 수 있습니다.

※ 아래 페이지가 출력된다면 Apache와 PHP가 정상적으로 연동된 것입니다.

PHP Version 5.6.18

System	Linux localhost.localdomain 3.10.0-327.13.1.el7.x86_64 #1 SMP Thu Mar 31 16:04:38 UTC 2016 x86_64
Build Date	May 4 2016 08:36:37
Configure Command	'./configure' '--with-mysql=/usr/local/mysql/bin/mysql_config,/configure' '--prefix=/usr/local/php' '--with-openssl=/usr/local/openssl' '--with-config-file-path=/usr/local/apache/conf' '--with-mysql=/usr/local/mysql' '--with-curl=/usr/local/curl' '--enable-calendar' '--enable-shmop' '--enable-ftp' '--enable-sockets' '--disable-cgi' '--with-gd' '--with-png-dir=/usr/lib' '--with-zlib-dir' '--with-jpeg-dir=/usr/lib' '--with-freetype-dir=/usr/lib' '--with-iconv' '--enable-mbstring' '--with-mcrypt=/usr/local' '--with-openssl' '--with-mysql=/usr/local/mysql/bin/mysql_config
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/apache/conf
Loaded Configuration File	/usr/local/apache/conf/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib *, convert.iconv.*, mcrypt.*, mdecrypt.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
with Zend Guard Loader v3.3, Copyright (c) 1998-2014, by Zend Technologies

zendengine

Configuration
apache2handler

Apache Version	Apache/2.4.18 (Unix) PHP/5.6.18
Apache API Version	20120211

10 mysql 단어를 검색 후 내용을 확인하여 연동된 것을 확인합니다.

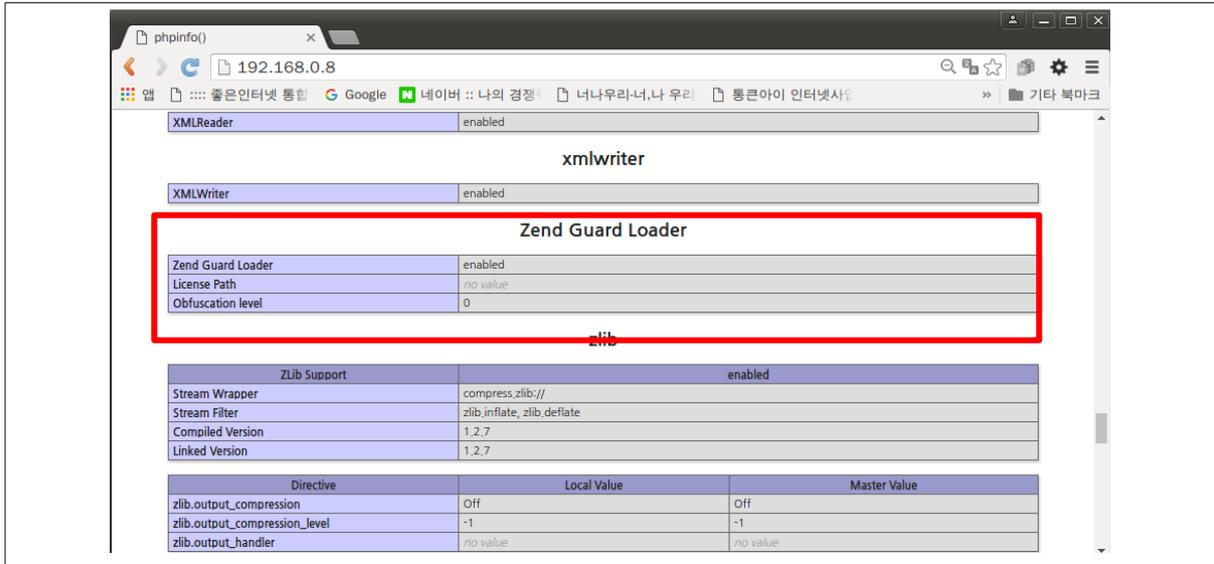
The screenshot shows the output of the phpinfo() function in a web browser. The browser's address bar shows '192.168.0.8'. The page title is 'phpinfo()'. The browser's address bar contains several tabs, including '좋은인터넷 통합', 'Google', '네이버 :: 나의 경쟁', '너나우리너,나 우리', and '통큰아이 인터넷사업'. The page content includes a table for 'Supported modes' and a table for 'mysql' support. The 'mysql' section is highlighted with a red box. Below it is the 'mysqli' section.

Directive	Local Value	Master Value
Supported modes	cbc cfb ctr ecb ncfb ofb stream	
mcrypt.algorithms_dir	no value	no value
mcrypt.modes_dir	no value	no value

mysql		
MySQL Support	enabled	
Active Persistent Links	0	
Active Links	0	
Client API version	5.6.10	
MYSQL_MODULE_TYPE	external	
MYSQL_SOCKET	/tmp/mysql.sock	
MYSQL_INCLUDE	-I/usr/local/mysql/include	
MYSQL_LIBS	-L/usr/local/mysql/lib -mysqlclient	
Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	/tmp/mysql.sock	/tmp/mysql.sock
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

mysqli		
Mysqli Support	enabled	
Client API library version	5.6.10	
Active Persistent Links	0	
Inactive Persistent Links	0	
Active Links	0	
Client API header version	5.6.10	
MYSQL_SOCKET	/tmp/mysql.sock	
Directive	Local Value	Master Value

11 Zend Guard loader 단어를 검색 후 내용을 확인하여 연동된 것을 확인합니다.



※ 참고 - phpinfo() 페이지의 설정 값 관련 참고 사이트.

<참고 URL>

<http://pmice.tistory.com/149>
http://phpschool.com/gnuboard4/bbs/board.php?bo_table=tipntech&wr_id=1452
<http://www.apmsetup.com/php.php?tutid=102&uid=139#139>
<http://php.net/manual/kr/ini.core.php>
<http://blog.naver.com/autovan74?Redirect=Log&logNo=100021168844>
<http://blog.missflash.com/561>

<공식 사이트 URL>

<http://www.php.net/manual/kr/ini.list.php>

3. 웹 서버 구축하기.

3.1 httpd.conf 환경 설정.

- 1 아파치의 주 설정파일인 httpd.conf 파일의 내용은 크게 3개의 section으로 분류됩니다. 아래 명령어를 실행하여 httpd.conf 설정을 확인합니다.
 - ➔ `vim /usr/local/apache/conf/httpd.conf`
- 2 Global Environment 에서 따로 설정할 값은 없고 아래 설정을 확인합니다.
 - **ServerRoot** : 아파치가 설치된 디렉토리 위치.
 - **Listen 80** : 아파치가 사용할 포트. 여러 개 지정 가능.
 - **LoadModule** : 아파치 실행시에 필요로 하는 기능들을 가진 파일을 불러오는 기능을 수행.
 - **User, Group** : 아파치 데몬은 root권한으로 실행하지만 자식 프로세스들은 이곳에서 지정한 User와 Group로 생성.

```
root@localhost:~  
ServerRoot "/usr/local/apache2"  
#  
# Listen: Allows you to bind Apache to specific IP addresses and/or  
# ports, instead of the default. See also the <VirtualHost>  
# directive.  
#  
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from gloomng onto all bound IP addresses.  
#  
Listen 12.34.56.78:80  
Listen 80  
#  
# Dynamic Shared Object (DSO) Support  
#  
# To be able to use the functionality of a module which was built as a DSO you  
# have to place corresponding 'LoadModule' lines at this location so the  
# directives contained in it are actually available _before_ they are used.  
# Statically compiled modules (those listed by 'httpd -l') do not need  
# to be loaded here.  
#  
# Example:  
# LoadModule foo_module modules/mod_foo.so  
LoadModule php5_module modules/libphp5.so  
LoadModule redurl_module modules/mod_url.so  
#  
<IfModule !mpm_netware_module>  
<IfModule !mpm_winnt_module>  
#  
# If you wish httpd to run as a different user or group, you must run  
# httpd as root initially and it will switch.  
#  
# User/Group: The name (or #number) of the user/group to run httpd as.  
# It is usually good practice to create a dedicated user and group for  
# running httpd, as with most system services.  
#  
User daemon  
Group daemon  
#  
</IfModule>  
</IfModule>  
72.0-1 7%
```

3 Main Server Configuration 에서 다음 설정 값을 적절히 수정합니다.

- **ServerAdmin** : 서버 관리자의 E-Mail 설정. (필요한 경우 변경.)
- **ServerName** : 접속할 주 서버의 이름. 도메인이 있다면 도메인 주소를 입력하고 없으면 IP 주소를 입력. (가상호스트를 사용하지 않을 경우 사용할 도메인 주소로 설정.)
- **DocumentRoot** : 클라이언트에서 ServerName으로 접속했을 때 보여지는 페이지가 저장된 디렉토리를 지정. (가상호스트를 사용하지 않을 경우 설정.)

```
root@localhost:~  
# 'Main' server configuration  
#  
# The directives in this section set up the values used by the 'main'  
# server, which responds to any requests that aren't handled by a  
# <VirtualHost> definition. These values also provide defaults for  
# any <VirtualHost> containers you may define later in the file.  
#  
# All of these directives may appear inside <VirtualHost> containers,  
# in which case these default settings will be overridden for the  
# virtual host being defined.  
#  
#  
# ServerAdmin: Your address, where problems with the server should be  
# e-mailed. This address appears on some server-generated pages, such  
# as error documents. e.g. admin@your-domain.com  
ServerAdmin you@example.com  
#  
# ServerName gives the name and port that the server uses to identify itself.  
# This can often be determined automatically, but we recommend you specify  
# it explicitly to prevent problems during startup.  
#  
# If your host doesn't have a registered DNS name, enter its IP address here.  
ServerName 127.0.1.1  
#  
# DocumentRoot: The directory out of which you will serve your  
# documents. By default, all requests are taken from this directory, but  
# symbolic links and aliases may be used to point to other locations.  
DocumentRoot "/usr/local/apache2/htdocs"  
#  
# Each directory to which Apache has access can be configured with respect  
# to which services and features are allowed and/or disabled in that  
# directory (and its subdirectories).  
#  
# First, we configure the "default" to be a very restrictive set of  
# features.  
115.1 18%
```

- **<Directory> ~ </Directory>** : 지정한 디렉토리를 어떤 방식으로 제어할 것인지 설정. (보안상 필요한 경우 설정.)

```
root@localhost:~  
<Directory />  
Options FollowSymLinks  
AllowOverride None  
Order deny,allow  
allow from all  
</Directory>  
123.1 27%
```

- **DirectoryIndex** : DocumentRoot에서 지정한 디렉토리에서 찾게 될 문서의 순서를 지정. (가상 호스트에도 같은 설정이 적용되므로 필히 설정)

Ex) 아래 그림처럼 두 개 이상의 파일을 지정했다면 홈페이지를 접속했을 때 index.html 파일을 먼저 찾아 본 후 없으면 index.php 파일을 참조하여 페이지를 찾아 로딩.

- **ErrorLog** : 아파치 웹 서버의 에러관련 로그를 저장할 파일의 위치를 지정. 아래 그림처럼 설정한 경우 DocumentRoot로 설정된 디렉토리 하위의 logs/error_log로 저장.
(Virtual Hosts 설정에서 각 사이트에 대한 에러 로그 파일을 지정하지 않으면 이 파일에 저장)

```

root@localhost:~#
<IfModule dir_module>
  DirectoryIndex index.html index.php
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "^\.ht">
  Order allow,deny
  allow from all
  Satisfy All
</FilesMatch>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error_log"
  
```

※ 그 외 자세한 설정은 아래 사이트를 참고하시기 바랍니다.

<http://www.linux.co.kr/home/lecture/index.php?cateNo=2&secNo=13&theNo=37&leccode=325>

3.2 가상 호스트 설정.

1 가상호스트 설정 방식에는 2가지가 있습니다.

1. httpd.conf에서 직접 가상호스트 설정.
2. httpd.conf에서 "Include conf/extra/httpd-vhosts.conf" 을 입력하여 httpd-vhosts.conf 파일에서 가상호스트 설정을 담당하도록 설정.

여기서는 httpd.conf에서 직접 가상호스트를 설정하는 방법에 대해서 알아보겠습니다.

2 httpd.conf 설정을 연 후 마지막 줄에 아래와 같이 양식에 맞게 추가합니다.

Ex) www.test.com, test1234.com 이라는 도메인 2개를 운영하려는 경우.

➔ vim /usr/local/apache/conf/httpd.conf

```

NameVirtualHost *:80

<VirtualHost *:80>
ServerAdmin Webmaster@test.com
DocumentRoot "/home/test.com"
ServerName test.com
ServerAlias www.test.com m.test.com
ErrorLog "logs/test.com-error_log"
CustomLog "logs/test.com-access_log" common
</VirtualHost>

<VirtualHost *:80>
ServerAdmin Webmaster@test1234.com
DocumentRoot "/home/test1234.com"
ServerName test1234.com
  
```

```
ServerAlias www.test1234.com m.test1234.com
ErrorLog "logs/test1234.com-error_log"
CustomLog "logs/test1234.com-access_log" common
</VirtualHost>
```

※ **ServerAdmin** : 해당 서버의 관리자 전자우편

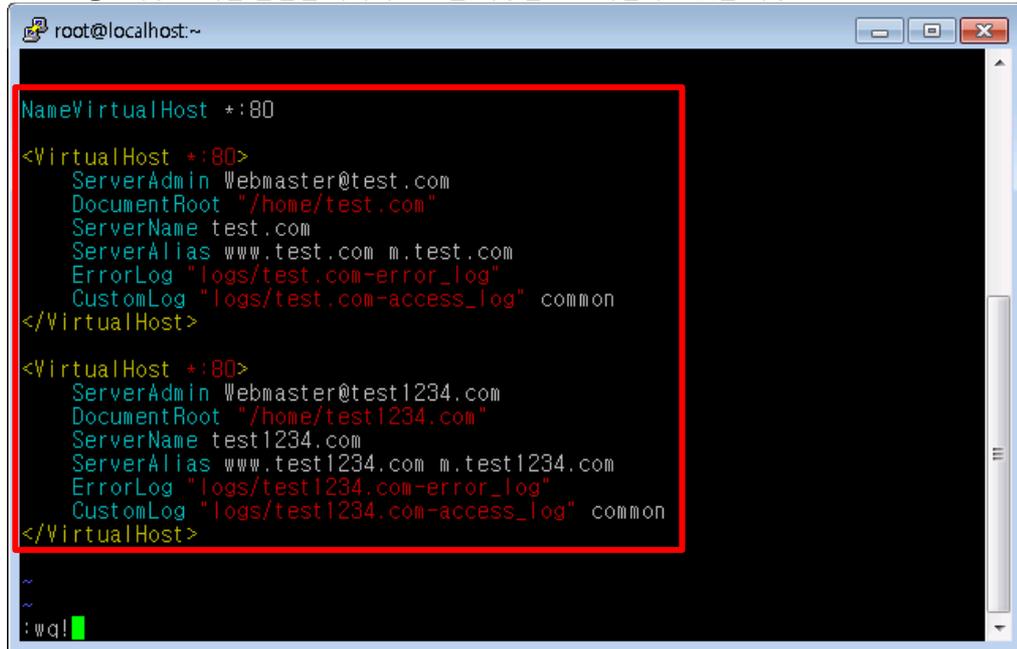
DocumentRoot : 해당 서버의 홈 디렉토리.

ServerName : 해당 서버의 도메인 주소.

ServerAlias : ServerName으로 등록된 도메인 주소로 연결할 다른 도메인 주소.

ErrorLog : 해당 도메인 관련된 아파치 에러가 발생하면 저장할 에러파일 지정.

CustomLog : 해당 도메인 관련된 아파치 로그를 저장할 로그파일과 포맷을 지정.



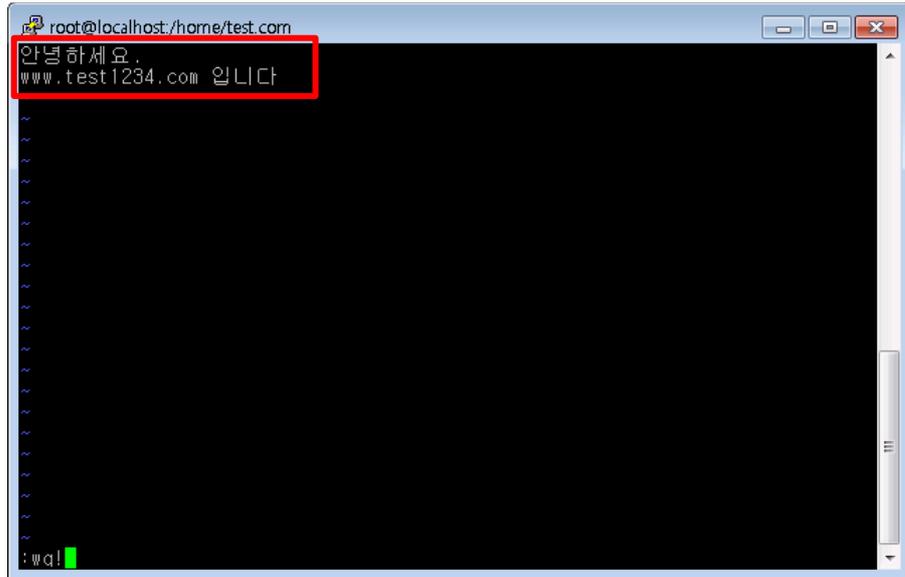
```
root@localhost:~#
NameVirtualHost *:80

<VirtualHost *:80>
  ServerAdmin Webmaster@test.com
  DocumentRoot "/home/test.com"
  ServerName test.com
  ServerAlias www.test.com m.test.com
  ErrorLog "logs/test.com-error_log"
  CustomLog "logs/test.com-access_log" common
</VirtualHost>

<VirtualHost *:80>
  ServerAdmin Webmaster@test1234.com
  DocumentRoot "/home/test1234.com"
  ServerName test1234.com
  ServerAlias www.test1234.com m.test1234.com
  ErrorLog "logs/test1234.com-error_log"
  CustomLog "logs/test1234.com-access_log" common
</VirtualHost>

~
~
:wq!
```

- 3 “:wq!”를 입력하여 설정을 저장한 후, 설정을 적용하기 위해 아파치를 재시작 합니다.
➔ service httpd restart

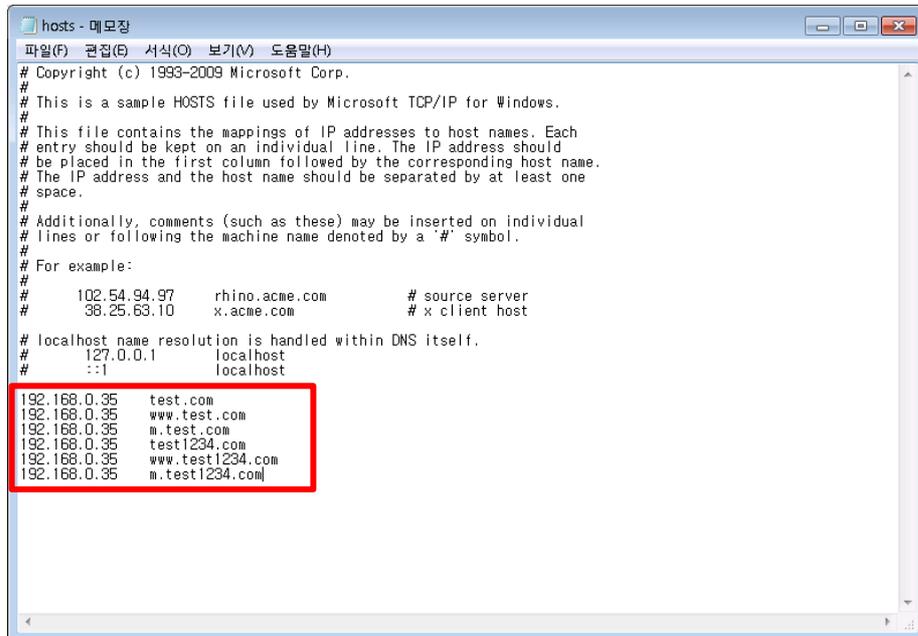


- 5 페이지가 정상적으로 출력되는지 외부에서 확인하기 위해 외부 PC의 hosts파일(DNS 서버보다 먼저 도메인주소와 IP를 확인하는 파일)을 메모장으로 열어, 하단에 아래 그림과 같이 추가하고 저장합니다.

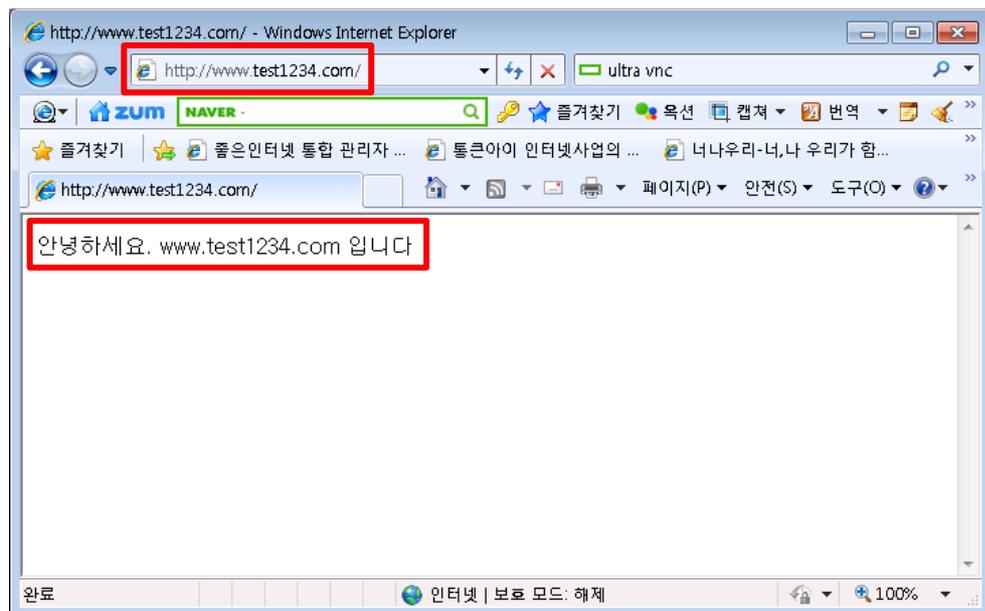
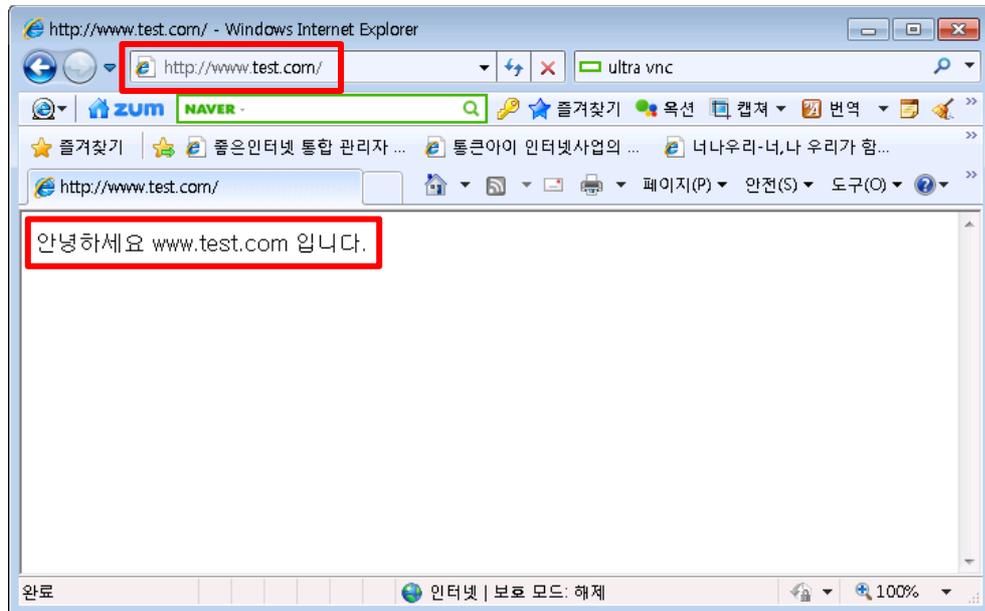
(테스트 환경: Windows 7, 8, 8.1 PC , 파일경로 C:\windows\system32\drivers\etc\hosts

※ 메모장으로 수정 안될 경우 : <http://pastimelife.com/229> 참고하여 관리자권한으로 수정 진행)

- ➔ <웹 서버 IP> test.com
- <웹 서버 IP> www.test.com
- <웹 서버 IP> m.test.com
- <웹 서버 IP> test1234.com
- <웹 서버 IP> www.test1234.com
- <웹 서버 IP> m.test1234.com



- 6 hosts파일을 편집한 테스트 PC에서 인터넷 브라우저 창을 열고 해당 웹 사이트 주소를 각각 입력하여 정상적으로 접속되는지 확인합니다.

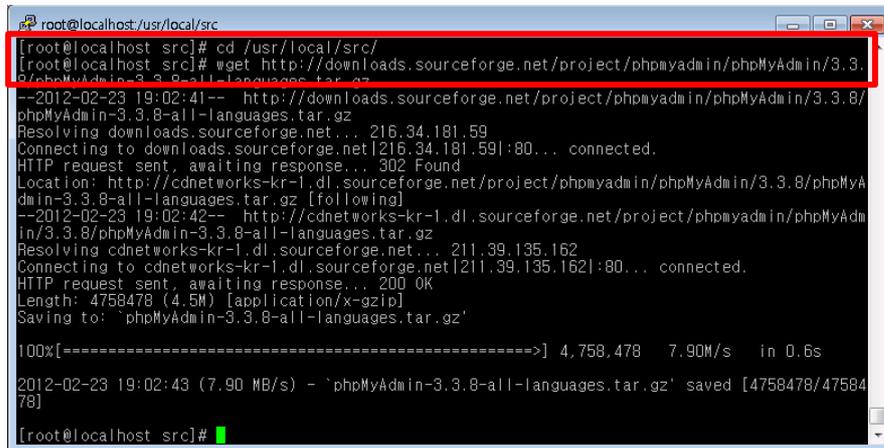


- 7 정상적으로 출력이 된다면 실제 적용할 도메인을 httpd.conf 설정에 적절히 적용시키고, DNS서버에서 해당 도메인에 대한 IP를 웹 서버의 IP로 변경하시면 됩니다.

4. 활용 하기.

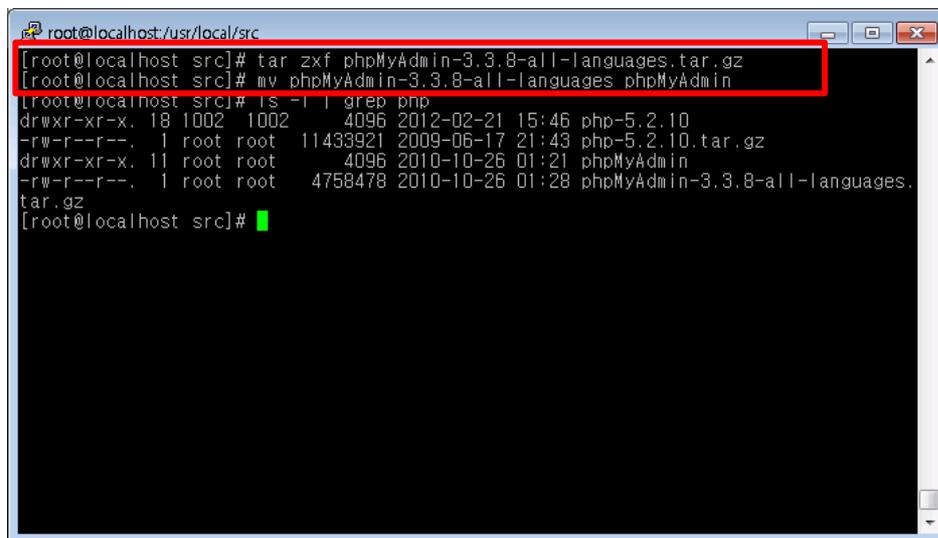
4.1 phpMyAdmin 설치하기.

- 1 phpMyAdmin은 MySQL을 웹에서 쉽게 관리할 수 있도록 GUI환경을 제공해주는 툴입니다.
/usr/local/src 에서 wget으로 phpMyAdmin 설치 파일을 다운 받습니다.
→ cd /usr/local/src
→ wget https://files.phpmyadmin.net/phpMyAdmin/4.5.3/phpMyAdmin-4.5.3-all-languages.zip



```
root@localhost:usr/local/src
[root@localhost src]# cd /usr/local/src/
[root@localhost src]# wget http://downloads.sourceforge.net/project/phpmyadmin/phpMyAdmin/3.3.8/phpMyAdmin-3.3.8-all-languages.tar.gz
--2012-02-23 19:02:41-- http://downloads.sourceforge.net/project/phpmyadmin/phpMyAdmin/3.3.8/
phpMyAdmin-3.3.8-all-languages.tar.gz
Resolving downloads.sourceforge.net... 216.34.181.59
Connecting to downloads.sourceforge.net[216.34.181.59]:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://cdnetworks-kr-1.dl.sourceforge.net/project/phpmyadmin/phpMyAdmin/3.3.8/phpMyAdmin-3.3.8-all-languages.tar.gz [following]
--2012-02-23 19:02:42-- http://cdnetworks-kr-1.dl.sourceforge.net/project/phpmyadmin/phpMyAdmin/3.3.8/phpMyAdmin-3.3.8-all-languages.tar.gz
Resolving cdnetworks-kr-1.dl.sourceforge.net... 211.39.135.162
Connecting to cdnetworks-kr-1.dl.sourceforge.net[211.39.135.162]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4758478 (4.5M) [application/x-gzip]
Saving to: `phpMyAdmin-3.3.8-all-languages.tar.gz'
100%[=====] 4,758,478 7.90M/s in 0.6s
2012-02-23 19:02:43 (7.90 MB/s) - `phpMyAdmin-3.3.8-all-languages.tar.gz' saved [4758478/4758478]
[root@localhost src]#
```

- 2 다운 받은 설치 파일의 압축을 풀고 폴더의 이름을 phpMyAdmin 으로 변경합니다.
→ unzip phpMyAdmin-4.5.3-all-languages.zip
→ mv phpMyAdmin-4.5.3-all-languages phpMyAdmin



```
root@localhost:usr/local/src
[root@localhost src]# tar xzf phpMyAdmin-3.3.8-all-languages.tar.gz
[root@localhost src]# mv phpMyAdmin-3.3.8-all-languages phpMyAdmin
[root@localhost src]# ls -l | grep php
drwxr-xr-x, 18 1002 1002 4096 2012-02-21 15:46 php-5.2.10
-rw-r--r--, 1 root root 11433921 2009-06-17 21:43 php-5.2.10.tar.gz
drwxr-xr-x, 11 root root 4096 2010-10-26 01:21 phpMyAdmin
-rw-r--r--, 1 root root 4758478 2010-10-26 01:28 phpMyAdmin-3.3.8-all-languages.tar.gz
[root@localhost src]#
```

- 3 phpMyAdmin을 /home 디렉토리로 복사하고, 도메인명 뒤에 phpMyAdmin을 입력하면 모두 phpMyAdmin 디렉토리를 접근할 수 있도록 httpd.conf에 Alias 설정을 추가합니다. 또한 phpMyAdmin 디렉토리의 index.php 파일을 인식할 수 있도록 설정에 index.php를 추가하고 설정이 적용되도록 아파치를 재 시작 합니다.
→ \cp -r /usr/local/src/phpMyAdmin /home/
→ vim /usr/local/apache/conf/httpd.conf

Alias /phpMyAdmin "/home/phpMyAdmin"

Alias /phpmyadmin "/home/phpMyAdmin"

DirectoryIndex index.html **index.php**

→ service httpd restart

```
root@localhost:/home
#
#   starting without SSL on platforms with no /dev/random equivalent
#   but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
<IfModule mod_url.c>
CheckURL On
</IfModule>
Alias /phpMyAdmin "/home/phpMyAdmin"
Alias /phpmyadmin "/home/phpMyAdmin"
Alias /userdb "/home/phpMyAdmin"

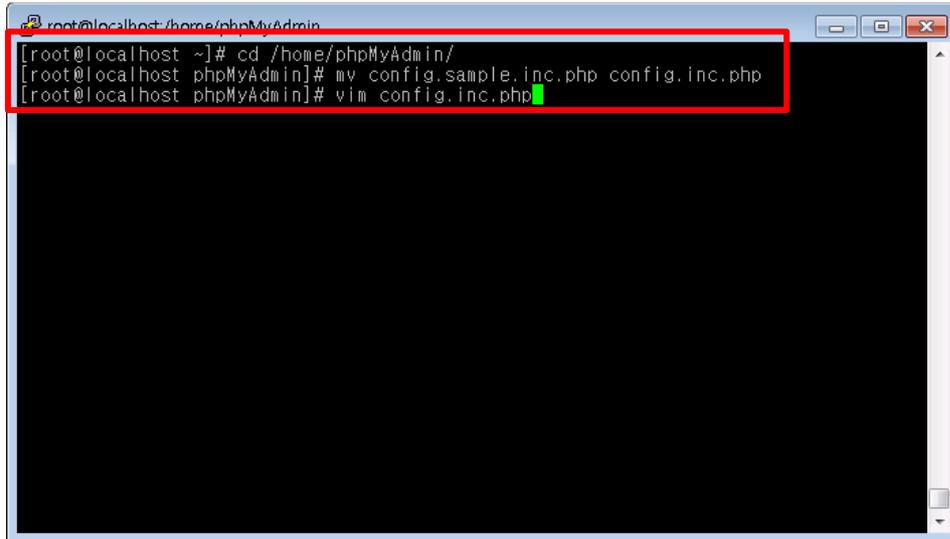
NameVirtualHost *:80

<VirtualHost *:80>
ServerAdmin Webmaster@test.com
DocumentRoot "/home/test.com"
432,1 96%
```

```
root@localhost:/home
#
#   Order allow,deny
#   Allow from all
#
</Directory>
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
DirectoryIndex index.html index.php
</IfModule>
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "^\.ht">
Order allow,deny
allow from all
Satisfy All
</FilesMatch>
176,20 36%
```

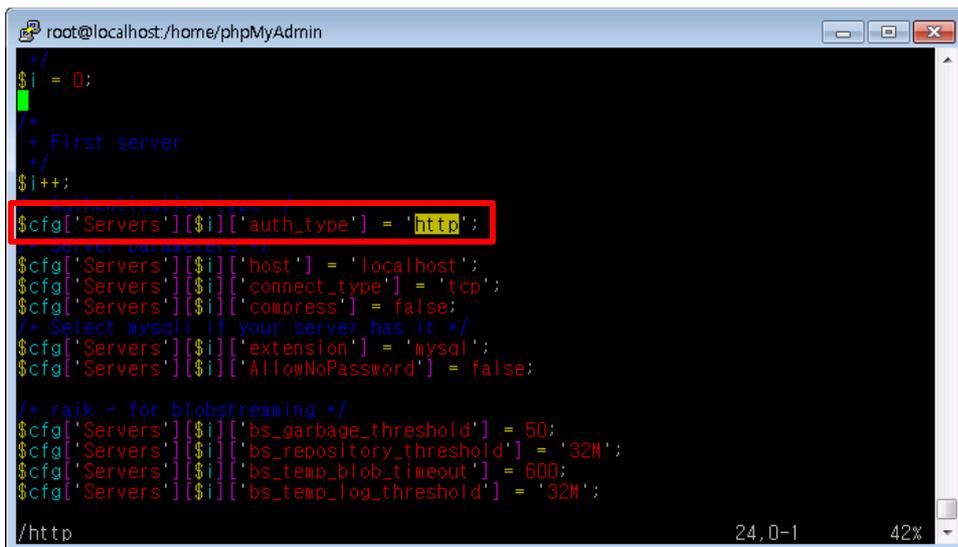
4 phpMyAdmin 폴더로 접근 후 샘플로 제공된 config 파일을 config.inc.php파일로 이름을 변경하고 해당 설정 파일을 편집합니다.

- cd /home/phpMyAdmin/
- mv config.sample.inc.php config.inc.php
- vim config.inc.php



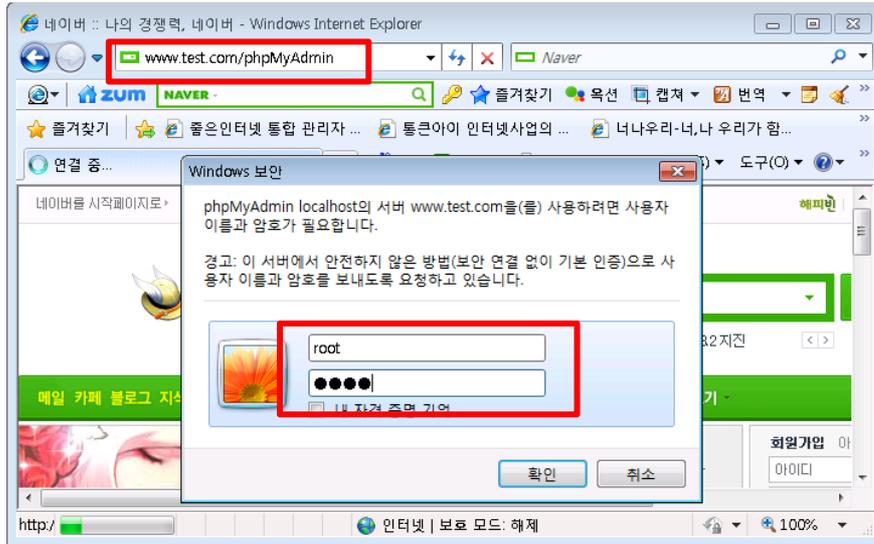
```
root@localhost /home/phpMyAdmin
[root@localhost ~]# cd /home/phpMyAdmin/
[root@localhost phpMyAdmin]# mv config.sample.inc.php config.inc.php
[root@localhost phpMyAdmin]# vim config.inc.php
```

- 5 인증 방식을 cookie로 이용할 경우 보안상 좋지 않으므로 auth_type 부분을 cookie에서 http로 변경하고 “:wq!”를 입력하여 저장합니다.

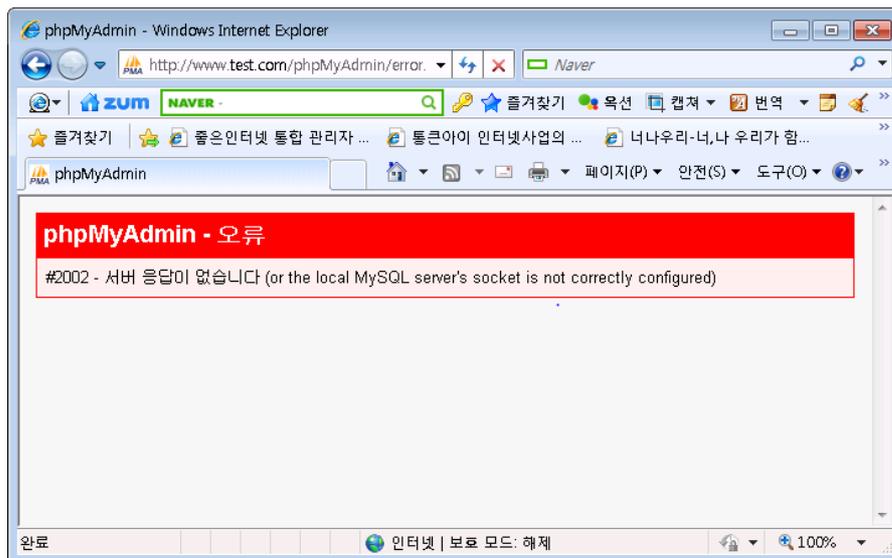


```
root@localhost /home/phpMyAdmin
+
$i = 0;
+ First server
+i++
+ authentication type +
$cfg['Servers'][$i]['auth_type'] = 'http';
+ server parameters +
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
+ Select mysql if your server has it +
$cfg['Servers'][$i]['extension'] = 'mysql';
$cfg['Servers'][$i]['AllowNoPassword'] = false;
+ rajk - for blobstreaming +
$cfg['Servers'][$i]['bs_garbage_threshold'] = 50;
$cfg['Servers'][$i]['bs_repository_threshold'] = '32M';
$cfg['Servers'][$i]['bs_temp_blob_timeout'] = 600;
$cfg['Servers'][$i]['bs_temp_log_threshold'] = '32M';
/htp 24,0-1 42%
```

- 6 외부PC에서 인터넷 브라우저 창을 열어 “http://도메인주소/phpMyAdmin” 을 입력하여 phpMyAdmin에 접속 후 팝업된 로그인 창에 MySQL 계정 ID와 패스워드를 입력합니다. (패스워드가 설정이 안 되어 있을 경우 접속이 안되므로 패스워드 설정 후 접속.)
→ Ex) www.test.com/phpMyAdmin



- 7 서버 환경에 따라 아래 화면처럼 접속 에러가 발생할 수 있는데, DB 호스트주소가 틀린 경우이므로 config.inc.php 파일의 host 부분을 localhost -> 127.0.0.1 혹은 localhost -> DB호스트주소로 수정하면 대부분 해결됩니다.



```

root@localhost:/home/phpMyAdmin
/*
+ First server
+ */
$i++
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'http';
$cfg['Servers'][$i]['host'] = '127.0.0.1';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
/* Select mysqli if your server has it */
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = false;

/* rawk - for blobstreaming */
$cfg['Servers'][$i]['bs_garbage_threshold'] = 50;
$cfg['Servers'][$i]['bs_repository_threshold'] = '32M';
$cfg['Servers'][$i]['bs_temp_blob_timeout'] = 600;
$cfg['Servers'][$i]['bs_temp_log_threshold'] = '32M';

/* User for advanced features */
// $cfg['Servers'][$i]['controluser'] = 'pma';
// $cfg['Servers'][$i]['controlpass'] = 'pmapass';
끝까지 찾았음, 처음부터 계속
28.5 48%

```

8 다시 한 번 접속하면 정상적으로 접속되는 것을 확인할 수 있습니다.



4.2 Log Rotate 설정.

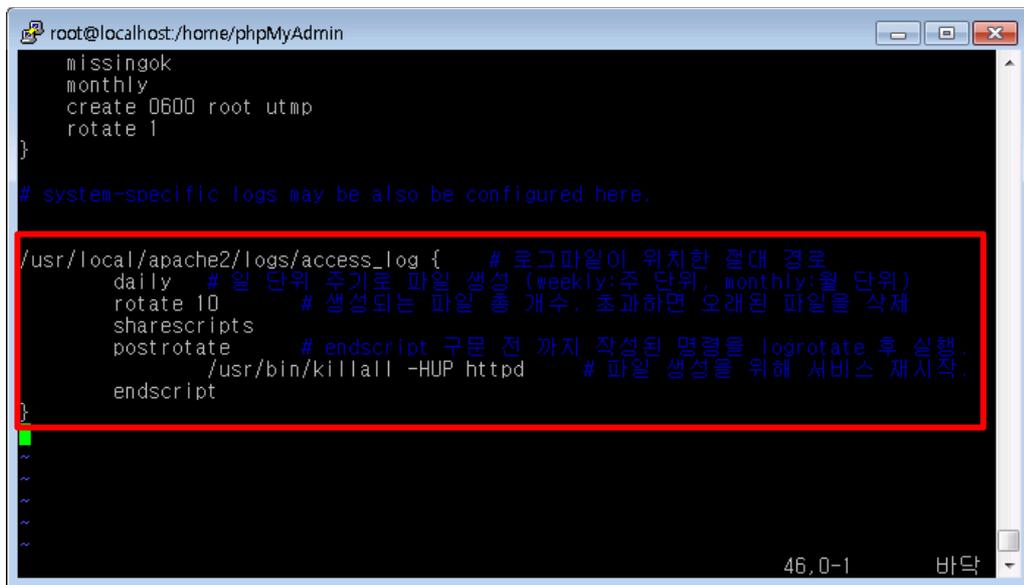
- 1 아파치 웹 서버를 운영하다 보면 접속자가 많을 경우 로그가 많이 쌓여 로그파일의 용량이 2GB가 되면 문제가 생길 수 있습니다. Log Rotate를 이용하여 일정 조건을 만족할 경우 새로운 로그파일을 생성하여 문제를 해결하도록 설정해야 합니다.
Log Rotate 설정을 담당하는 /etc/logrotate.conf 파일을 편집하여 아래 양식대로 추가하고 저장

합니다. (설정 값은 환경에 따라 적절히 변경하고, 파란색으로 표기된 주석부분은 실제 적용 시 문제가 되기 때문에 삭제하시기 바랍니다.)

→ vim /etc/logrotate.conf

```
/usr/local/apache/logs/access_log { # 로그파일이 위치한 절대 경로
    daily # 일 단위 주기로 파일 생성 (weekly:주 단위, monthly:월 단위)
    rotate 10 # 생성되는 파일 총 개수. 초과하면 오래된 파일을 삭제
    sharedscripts
    postrotate # endsript 구문 전 까지 작성된 명령을 logrotate 후 실행.
                /usr/bin/killall -HUP httpd # 파일 생성을 위해 서비스 재시작.
    endsript
}
```

:wq!



```
root@localhost:/home/phpMyAdmin
missingok
monthly
create 0600 root utmp
rotate 1
}
# system-specific logs may be also be configured here.

/usr/local/apache2/logs/access_log { # 로그파일이 위치한 절대 경로
    daily # 일 단위 주기로 파일 생성 (weekly:주 단위, monthly:월 단위)
    rotate 10 # 생성되는 파일 총 개수. 초과하면 오래된 파일을 삭제
    sharedscripts
    postrotate # endsript 구문 전 까지 작성된 명령을 logrotate 후 실행.
                /usr/bin/killall -HUP httpd # 파일 생성을 위해 서비스 재시작.
    endsript
}
~
~
~
46,0-1 바닥
```

2 설정한 Log Rotate를 반복 예약작업에 등록하기 위해 /etc/crontab을 확인합니다. 아래 화면처럼 run-parts가 존재한다면 별도의 작업이 필요 없지만 없는 경우 아래 화면처럼 추가해주어야 합니다.

→ vim /etc/crontab

```

root@localhost:/etc/cron.daily
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly

# For details see man 4 crontabs

# Example of job definition:
#----- minute (0 - 59)
# |----- hour (0 - 23)
# | |----- day of month (1 - 31)
# | | |----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | |----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,
Sat
# | | | | |
# * * * * * user-name command to be executed
~
21.1 모두

```

3 /etc/cron.daily 에 logrotate.conf에 설정된 내용을 실행하는 logrotate 라는 스크립트 파일이 있는지 확인합니다.

➔ ls -al /etc/cron.daily/

```

root@localhost:~
[root@localhost ~]# ls -al /etc/cron.daily/
합계 32
drwxr-xr-x. 2 root root 4096 2012-02-24 17:33 .
drwxr-xr-x. 93 root root 4096 2012-02-24 18:05 ..
-rwxr-xr-x. 1 root root 196 2011-06-25 20:17 logrotate
-rwxr-xr-x. 1 root root 905 2010-11-12 05:45 makepasswd.cron
-rwxr-xr-x. 1 root root 174 2009-05-20 15:03 mlocate.cron
-rwxr-xr-x. 1 root root 2126 2010-04-23 23:34 prelink
-rwxr-xr-x. 1 root root 563 2010-08-23 18:35 readahead.cron
-rwxr-xr-x. 1 root root 365 2009-10-16 14:52 tmpwatch
[root@localhost ~]#

```

4 Log Rotate로 설정한 주기(Ex: 하루)가 지난 후 로그파일이 제대로 생성되었는지 확인합니다.

➔ ls -al /usr/local/apache/logs

```

root@localhost:~# ls -al /usr/local/apache2/logs/
합계 76
drwxr-xr-x. 2 root root 4096 2012-02-24 17:36 .
drwxr-xr-x. 15 root root 4096 2012-02-21 15:39 ..
-rw-r--r--. 1 root root 0 2012-02-24 17:36 access_log
-rw-r--r--. 1 root root 1509 2012-02-23 14:40 access_log-20120224
-rw-r--r--. 1 root root 8482 2012-02-24 17:36 error_log
-rw-r--r--. 1 root root 5 2012-02-24 17:36 httpd.pid
-rw-r--r--. 1 root root 33973 2012-02-24 15:03 test.com-access_log
-rw-r--r--. 1 root root 1600 2012-02-23 19:20 test.com-error_log
-rw-r--r--. 1 root root 1519 2012-02-23 19:22 test1234.com-access_log
-rw-r--r--. 1 root root 543 2012-02-23 17:57 test1234.com-error_log
[root@localhost ~]#

```

4.3 SSL 보안 서버 설치하기.

- 1 Apache에 SSL 인증서를 설치하여 보안 서버를 구축 할 수 있습니다.
보안서버란 인터넷 상에서 사용자 PC와 웹 서버 사이에 송수신 되는 개인 정보를 암호화 하여 전송 하는 서버를 의미합니다.
- 2 SSL 인증서 발급 기관에 따라 설치방법에 대해 상이하므로 SSL 인증서 제공업체에 상세설정은 문의해보셔야 합니다.
- 3 설치방법
 - a. Apache conf 파일 수정

```

[root@localhost ~]# vim /usr/local/apache/conf/httpd.conf
- LoadModule ssl_module modules/mod_ssl.so의 주석 제거.

```

```

root@localhost:/usr/local/apache/conf
#LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_express_module modules/mod_proxy_express.so
#LoadModule session_module modules/mod_session.so
#LoadModule session_cookie_module modules/mod_session_cookie.so
#LoadModule session_dbd_module modules/mod_session_dbd.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
LoadModule ssl_module modules/mod_ssl.so
#LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
#LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
#LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
#LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule unixd_module modules/mod_unixd.so
#LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule info_module modules/mod_info.so
<IfModule mpm_prefork_module>
    #LoadModule cgi_module modules/mod_cgi.so
</IfModule>
"httpd.conf" 504L, 18193C 132,1 24%

```

- Secure (SSL/TLS)connections 부분 주석 제거.

```
# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

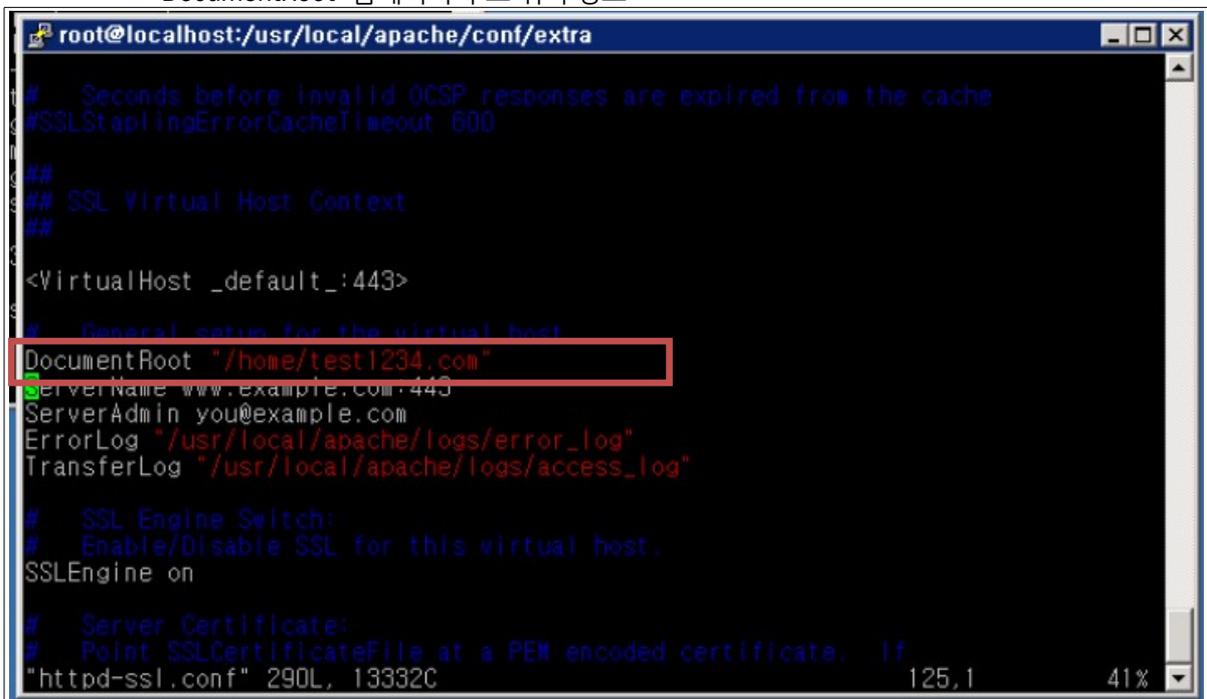
# Various default settings
#Include conf/extra/httpd-default.conf

# Secure (SSL/TLS) connections
include conf/extra/httpd-ssl.conf

#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
```

b. httpd-ssl.conf 파일 수정

- [root@localhost ~]# vim /usr/local/apache/conf/extra/httpd-ssl.conf
- DocumentRoot "홈페이지 루트 위치 경로"



```
root@localhost:~/usr/local/apache/conf/extra
# Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

##
## SSL Virtual Host Context
##
<VirtualHost _default_:443>
# General setup for the virtual host
DocumentRoot "/home/test1234.com"
ServerName www.example.com:443
ServerAdmin you@example.com
ErrorLog "/usr/local/apache/logs/error_log"
TransferLog "/usr/local/apache/logs/access_log"

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
"httpd-ssl.conf" 290L, 13332C 125,1 41%
```

- SSLCertificateFile “/etc/pki/tls/ca.crt” 으로 수정
- SSLCertificateKeyFile “/etc/pki/tls/ca.key” 으로 수정

```

root@localhost:/usr/local/apache/conf/extra
# In mind that if you have both an RSA and a DSA certificate you
# can configure both in parallel (to also allow the use of DSA
# ciphers, etc.)
# Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
# require an ECC certificate which can also be configured in
# parallel.
#SSLCertificateFile "/usr/local/apache/conf/server.crt"
SSLCertificateFile "/etc/pki/tls/ca.crt"
#SSLCertificateFile "/usr/local/apache/conf/server-dsa.crt"
#SSLCertificateFile "/usr/local/apache/conf/server-ecc.crt"
#
# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
# ECC keys, when in use, can also be configured in parallel.
SSLCertificateKeyFile "/etc/pki/tls/ca.key"
#SSLCertificateKeyFile "/usr/local/apache/conf/server-dsa.key"
#SSLCertificateKeyFile "/usr/local/apache/conf/server-ecc.key"
#
# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the

```

c. **SSL 보안서버 인증서 생성 및 신청**

(※ 각 SSL 보안서버 인증서 기관에 상의하므로 상세단계는 해당기관에 문의하시기 바랍니다.)

- CA용 인증키 생성

```
[root@localhost ~]# openssl genrsa -des3 -out ca.key 1024
```

L 암호입력 (※ 보안서버 인증서생성을 위해 필요하니 기억필수!!)

```

[root@localhost extra]# openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.+++++
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
[root@localhost extra]#

```

- CA용 인증서생성

```
[root@localhost ~]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

L CA용 인증키 입력 → 인증할 사이트 정보 입력 (나라, 회사명등)

```
[root@localhost extra]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:KR
State or Province Name (full name) []:seoul
Locality Name (eg, city) [Default City]:Seoul
Organization Name (eg, company) [Default Company Ltd]:Companyname
Organizational Unit Name (eg, section) []:CompanyUnitname
Common Name (eg, your name or your server's hostname) []:RHEL
Email Address []:test@test.com
```

아래와 같이 key와 crt 파일이 생성됨을 확인할 수 있음

```
[root@localhost extra]# ls
ca.crt          httpd-info.conf      httpd-ssl.conf
ca.key          httpd-languages.conf httpd-userdir.conf
httpd-autoindex.conf httpd-manual.conf    httpd-vhosts.conf
httpd-dav.conf  httpd-mpm.conf       proxy-html.conf
httpd-default.conf httpd-multilang-error.doc
```

- SSL 보안인증서 발급업체에 앞에서 생성한 CA용 인증서와 인증키, 생성시 사용된 사이트정보를 알려주고 CSR 파일을 기다린다.

- CA용 인증서와 인증키 파일과 발급업체로 받은 CSR파일을 /etc/pki/tls디렉토리로 복사한다

```
[root@localhost ~]# cp ca.crt /etc/pki/tls/certs
```

```
[root@localhost ~]# cp ca.key /etc/pki/tls/ca.key
```

```
[root@localhost ~]# cp ca.csr /etc/pki/tls/ca.csr {※ ca.csr은 발급업체에서 받은 파일}
```

d. **Apache conf** 파일 수정 (**ssl** 응답 가상호스트 설정)

```
[root@localhost ~]# vim /usr/local/apache/conf/httpd.conf
```

→ SSL 가상호스트 설정 추가

```
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/ca.crt
    SSLCertificateKeyFile /etc/pki/tls/private/ca.key
    ServerAdmin test1234.com
    ServerName test1234.com
    DocumentRoot /home/test1234.com
    CustomLog logs/ssl_starkapin_com_error_log common
    ErrorLog logs/ssl_starkapin_com_error_log
```

</VirtualHost>

```
<VirtualHost *:443>
  SSLEngine on
  SSLCertificateFile /etc/pki/tls/certs/ca.crt
  SSLCertificateKeyFile /etc/pki/tls/private/ca.key
  ServerAdmin test1234.com
  ServerName test1234.com
  DocumentRoot /home/test1234.com
  CustomLog logs/ssl_starkapin_com_error_log common
  ErrorLog logs/ssl_starkapin_com_error_log
</VirtualHost>
"httpd.conf" 516L, 18542C
```

4. 아파치를 재시작하면 보안서버가 적용된다
 - a. 재부팅시 설정한 CA 인증키값 패스워드를 넣어야지 사이트가 실행된다.

```
[root@localhost test]# service httpd restart
httpd 를 정지 중: [ OK ]
httpd (을)를 시작 중: Apache/2.2.2 mod_ssl/2.2.2 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

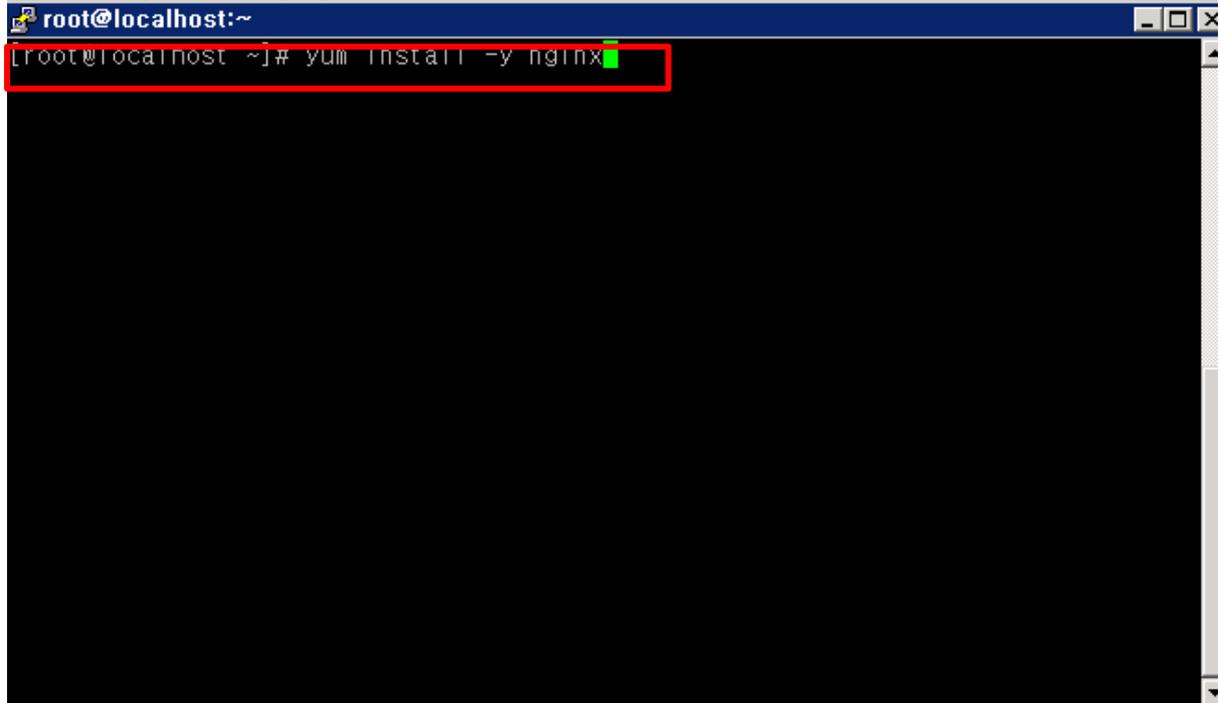
Server 192.168.0.28:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful. [ OK ]
[root@localhost test]# █
```

5. 다른 웹프로그램(nginx)을 이용하여 서버 구축하기.

5.1 nginx 설치

- 1 nginx은 apache와 동일하게 웹파일을 기동시주는 웹 서버 프로그램으로 최근 프록시기능, 로드밸런싱, 속도향상등 apache 기능보다 향상된 기능을 지원하여 사용자가 늘고 있습니다.
- 2 yum install을 통해 nginx 패키지를 설치합니다.
-> yum install -y nginx



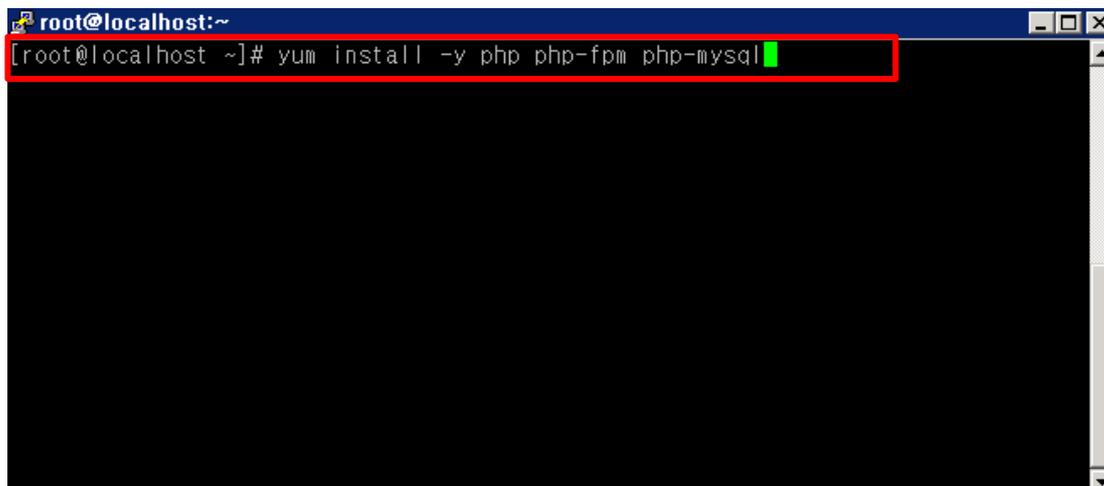
```
root@localhost:~  
[root@localhost ~]# yum install -y nginx
```

- 3 부팅시 자동시작하도록 설정.
-> systemctl enable nginx.service
- 4 서비스를 시작합니다
-> service nginx start
- 5 웹브라우저에 http://서버IP로 접속하여 nginx에 정상적으로 동작하는지 확인합니다.



6 php연동을 위해 php5-fpm을 설치 합니다.

=> yum install -y php php-fpm php-mysql



7 cgi.fix_pathinfo는 해당하는 파일이 없을 경우 가장근정합 파일을 실행시켜주는 기능으로 보안적으로 위험할 수 있어 비활성한다.

=> vim /etc/php.ini

php 설정파일을 연다

=> cgi.fix_pathinfo의 주석(; 제거)하고 1을 0으로 변경함

```

root@ubuntu: ~
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not g
rok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Set
ting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setti
ng
; of zero causes PHP to behave as before. Default is 1. You should fix your sc
ripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0
; FastCGI under IIS (on WINNT based OS) supports the ability to impersonate
; security tokens of the calling client. This allows IIS to define the
; security context that the request runs under. mod_fastcgi under Apache
; does not currently support this feature (03/17/2002)
; Set to 1 if running under IIS. Default is zero.
; http://php.net/fastcgi.impersonate
;fastcgi.impersonate = 1

; Disable logging through FastCGI connection. PHP's default behavior is to enabl
e
; this feature.

769,0-1 39%

```

- 8 php-fpm 설정을 변경하여 통신 sock 파일의 생성 패스를 설정한다
 - => vim /etc/php-fpm.d/www.conf
 - php-fpm 설정파일을 연다
 - => listen = /var/run/php-fpm.sock 으로 수정한다

```

root@localhost:~
; Start a new pool named 'www'.
[www]

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'   - to listen on a TCP socket to a specific address on
;                           a specific port;
;   'port'                - to listen on a TCP socket to all addresses on a
;                           specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /var/run/php-fpm.sock
; Set listen(2) backlog. A value of '-1' means unlimited.
; Default Value: -1
;listen.backlog = -1

"/etc/php-fpm.d/www.conf" 226L, 10025C 13,0-1 Top

```

- 9 php-fpm 을 재시작하고 부팅시에 자동시작할 수 있도록 설정한다
 - => service php-fpm restart
 - php-fpm 재시작한다
 - => systemctl enable php-fpm

5.2 nginx – php 연동

- 1 /etc/nginx/nginx.conf 을 수정하여 php를 연동할 수 있도록 수정한다

```
=> mv -f /etc/nginx/nginx.conf /etc/nginx/nginx.conf_bak
    기존파일을 백업하고,

=> vim /etc/nginx/nginx.conf
    설정파일을 수정하여 php를 연동할 수 있도록 구성한다 (빨간색부분만 추가하거나 수정한다)

server {
    listen    80 default_server;
    listen    [::]:80 default_server;
    server_name _;
    root      /usr/share/nginx/html;
    index index.php index.html index.htm;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        try_files $uri $uri/ =404;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }

    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

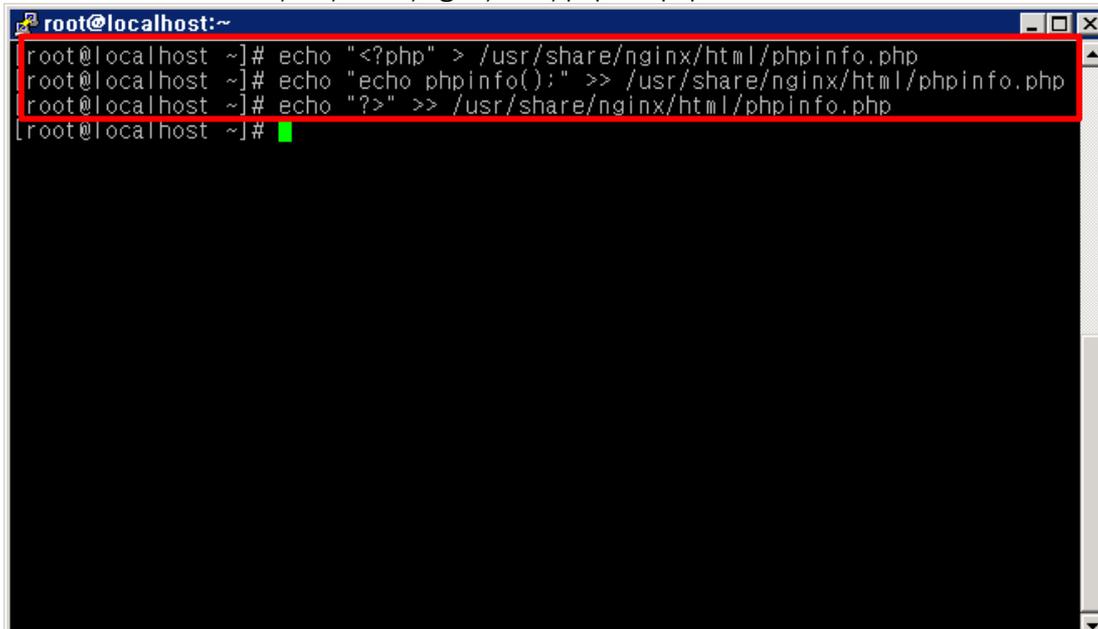

3 phpinfo 페이지 생성하여 정상연동여부를 확인함

=> 페이지생성

echo "<?php" > /usr/share/nginx/html/phpinfo.php

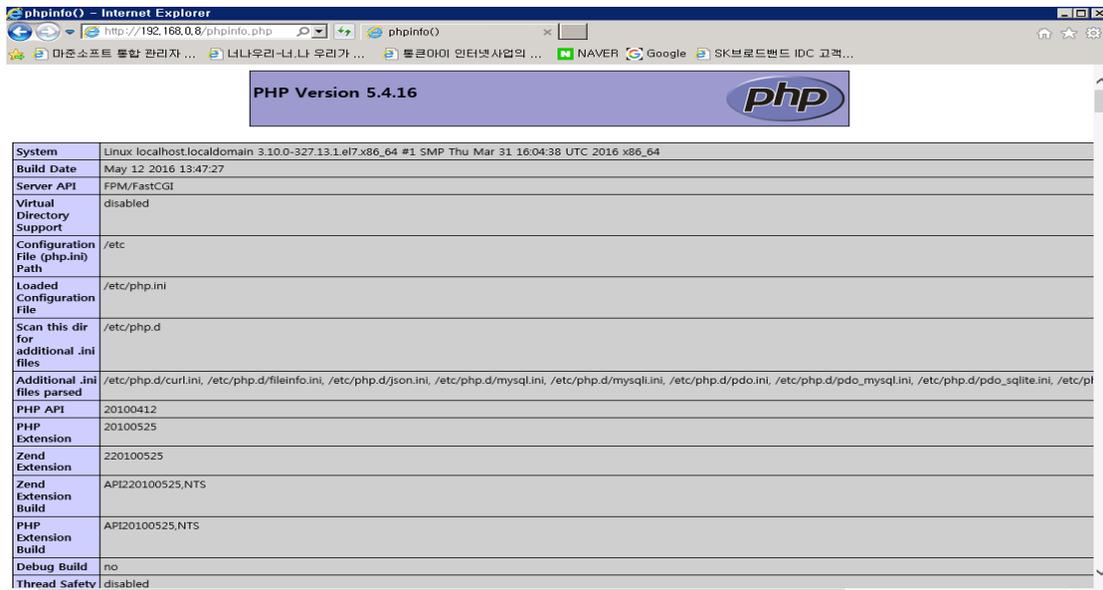
echo "echo phpinfo();" >> /usr/share/nginx/html/phpinfo.php

echo "?>" >> /usr/share/nginx/html/phpinfo.php



```
root@localhost:~# echo "<?php" > /usr/share/nginx/html/phpinfo.php
root@localhost:~# echo "echo phpinfo();" >> /usr/share/nginx/html/phpinfo.php
root@localhost:~# echo "?>" >> /usr/share/nginx/html/phpinfo.php
root@localhost:~#
```

=> 웹브라우저에서 php 페이지가 열리는지 여부 확인되면 정상연동된것이다.



=> 웹브라우저에서 php 페이지가 열리는지 여부 확인되면 정상연동된것이다.

5.3 nginx 설정

1 가상호스팅(복수도메인사이트등록) :

=> /etc/nginx/conf.d/{추가할 사이트명}.conf 파일을 생성하여 세팅함
필요한 만큼 파일을 생성하여 사이트를 추가하면 됩니다. (빨간색부분을 맞게 수정하십시오)

```
=> vim /etc/nginx/conf.d/test.com.conf
설정파일을 수정하여 호스팅을 추가한다

#도메인명 : www.test.com
server {
    listen 192 .168.8.43:80;

    root /home/test.com;
    index index.php index.html index.htm;

    access_log /var/log/test.com_access.log;
    server_name www.test.com;

    location / {
        try_files $uri $uri/ =404;
    }

    error_page 404 /404.html;
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

2 에러로그, 액세스로그 세팅 :

```
=> vim /etc/nginx/conf.d/test.com.conf
server {
    ~
    access_log /var/log/test.com_access.log main;
    error_log /var/log/test.com_error.log main;
    ~
}
```

3 초기 페이지 명, 사이트 Root 디렉토리 위치, 통신포트

```
=> vim /etc/nginx/conf.d/test.com.conf
server {
  ~
  location / {
    listen 80 ; # listen 포트번호
    root /home/test.com; # Root 디렉토리 경로;
    index index.php index.html; #index {초기 페이지 명};
  }
  ~
}
```

감사합니다.