

# 서버 기본 설정

SSH, FTP, APM

## Ubuntu

### [SSH](#) 접속방법



### FTP 서버 설치 및 접속

FTP 설치  
FTP 설정



### APM 설치

Apache 설치 MySQL 설치  
PHP 설정  
작동 확인

## CentOS

### [SSH](#) 설정



### FTP 서버 제작 및 접속

FTP 설치  
FTP 설정

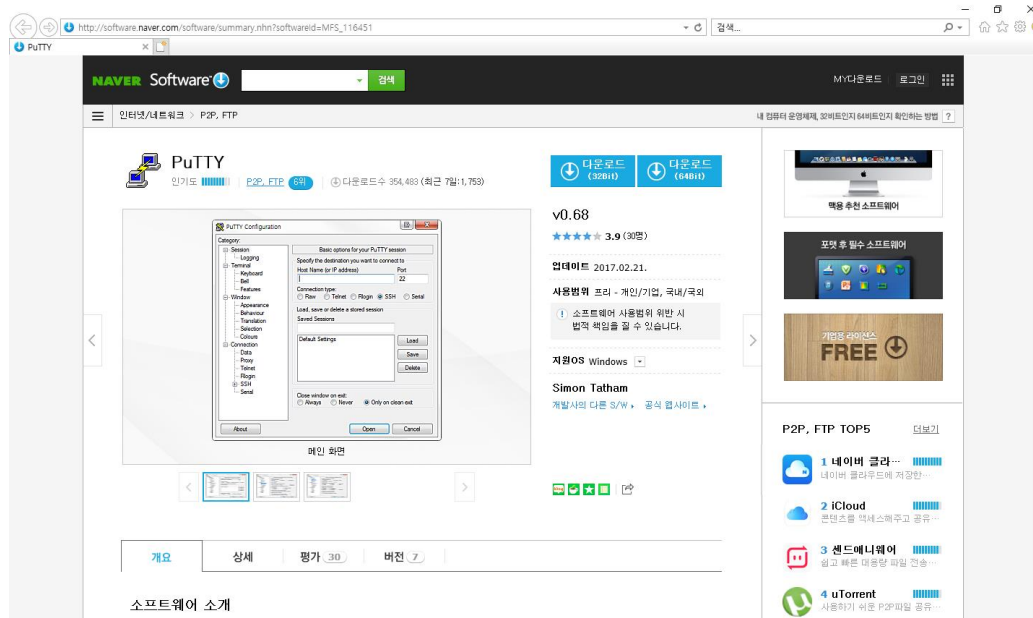


### APM 설치

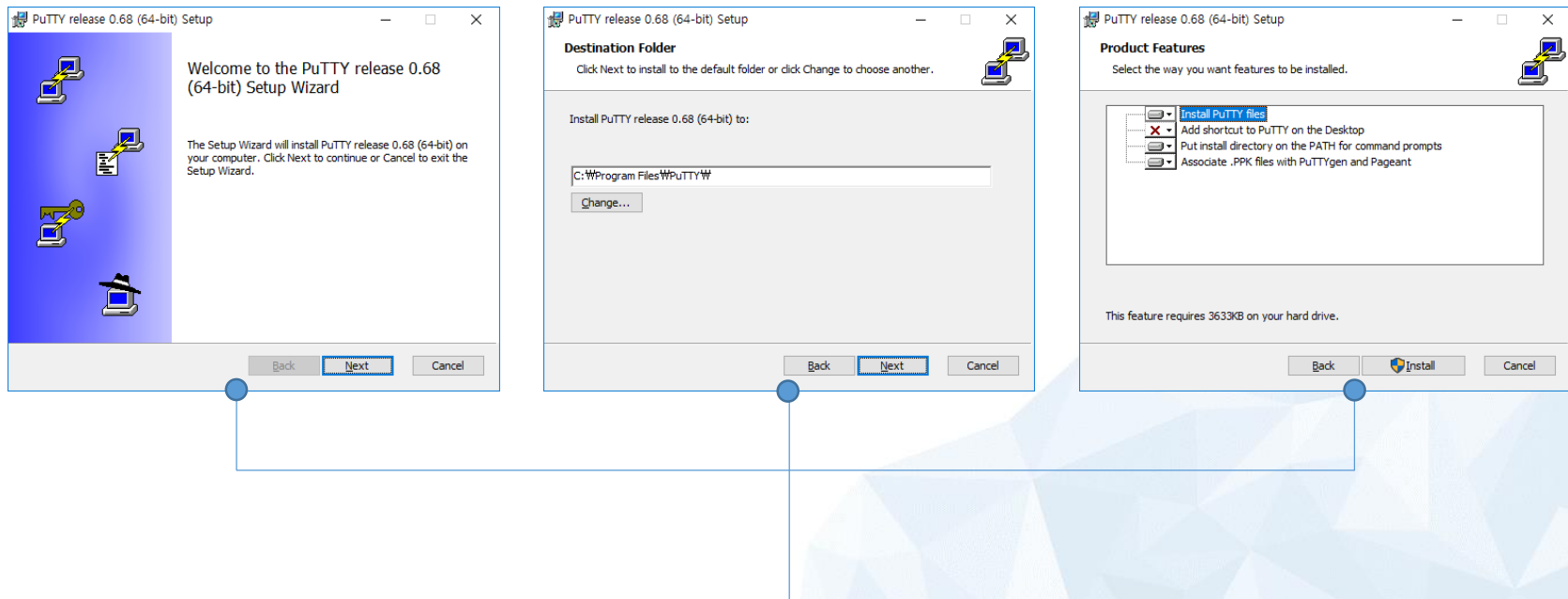
Yum 업데이트  
기능 설치  
작동 확인  
방화벽 설정

# SSH 접속 방법

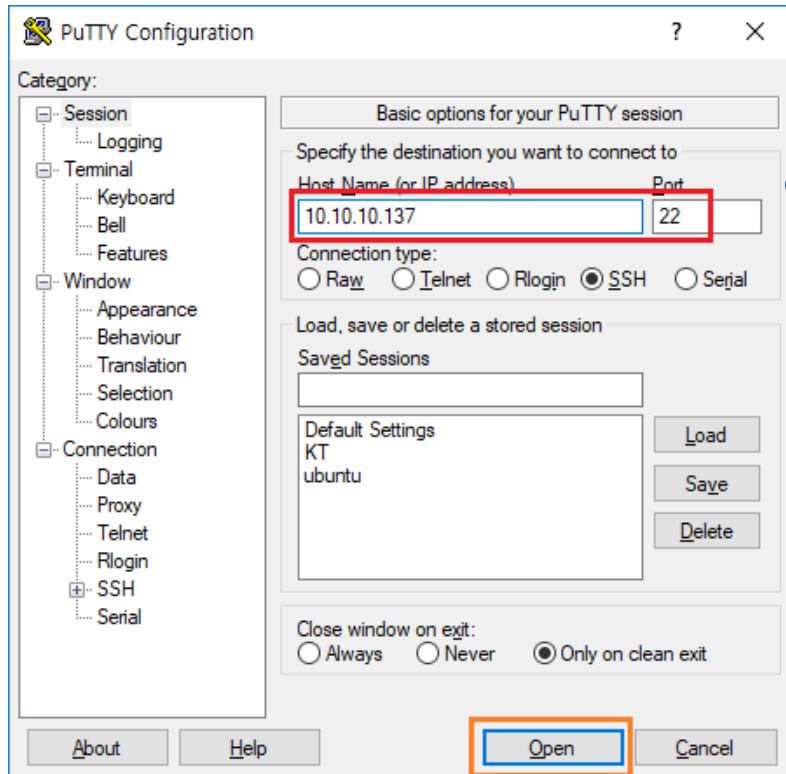
# PuTTY 설치



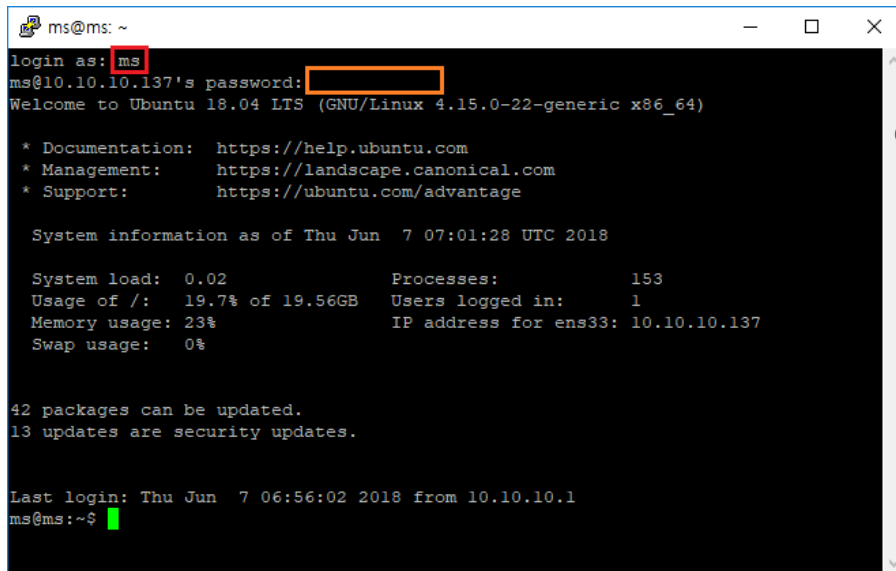
자신이 사용하는 PC에  
PuTTY를 설치합니다.



위와 같이 설치 과정을 거쳐 설치 해줍니다.



설치가 완료되었다면  
PuTTY를 실행하여 자신이 접속하려는  
아이피 주소와 포트를 명시한 후  
Open을 눌러줍니다.



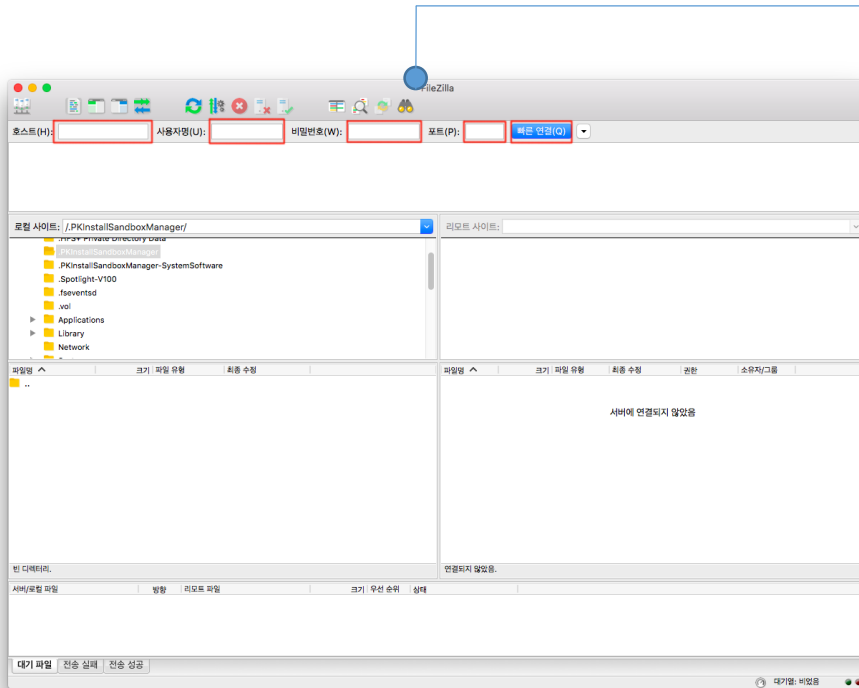
```
ms@ms: ~  
login as: ms  
ms@10.10.10.137's password:  
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Thu Jun  7 07:01:28 UTC 2018  
  
System load:  0.02          Processes:      153  
Usage of /:   19.7% of 19.56GB  Users logged in:  1  
Memory usage: 23%          IP address for ens33: 10.10.10.137  
Swap usage:   0%  
  
42 packages can be updated.  
13 updates are security updates.  
  
Last login: Thu Jun  7 06:56:02 2018 from 10.10.10.1  
ms@ms:~$
```

SSH 설정이 완료된 서버에 접속하게 되면  
사진과 같이 서버에  
정상적으로 접속이 됩니다.

# FTP 서버 설치 및 접속



# FTP 접속 방법



호스트에는 자신이 접속하는 FTP서버의  
아이피 사용자명은 자신이 접속하려는  
사용자 명 비밀번호는  
그 사용자의 비밀번호 포트는  
자신이 설정한 포트를 입력한 후  
빠른 접속을 눌러 접속합니다.

정상적으로 설정 되어있다면 접속이  
가능합니다.

# Ubuntu

sudo apt-get install vsftpd

(FTP 설치 명령어)



명령어 입력 후 비밀번호를 눌러줍니다.



그 후에 설치 진행을 물어보는데요.  
y를 눌러주면 진행 n을 누르면  
진행하지 않습니다.

```
kuj@kuj:~$ sudo apt-get install vsftpd  
[sudo] password for kuj:
```

\* “vs”는 very secure의 약자로 보안성의 취약점을 보완했기에 붙여진 이름입니다.

```
kuj@kuj:~$ sudo vi /etc/vsftpd.conf
```

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
"/etc/vsftpd.conf" 155L, 5850C
```

1,1

Top

FTP 연결 설정을 수정하기 위해  
“vi /etc/vsftpd.conf”  
명령어를 통해 접속하여줍니다.

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
"/etc/vsftpd.conf" 155L, 5850C
```

1,1

Top

우선 보안을 위해서 익명의 사용자는 vsftpd  
를 사용할 수 없도록 해야 합니다.

“anonymous\_enable=NO”란  
익명 계정 접속 불가라는 것인데  
허용해주고 싶다면 YES로 바꿔주면 됩니다.

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
"/etc/vsftpd.conf" 155L, 5850C
```

1,1

Top

파일 읽기 쓰기를가능하게 해야 하므로  
“listen=No” 부분을 YES로 바꿔주고  
“write\_enable=YES”부분의  
주석을 풀어줍니다.

수정은 끝났으니 ESC 후 :wq를  
입력하여 저장 후 종료해줍니다.

```
kuj@kuj:~$ sudo service vsftpd restart_
```

```
kuj@kuj:~$ sudo adduser testuser
Adding user `testuser' ...
Adding new group `testuser' (1005) ...
Adding new user `testuser' (1001) with group `testuser' ...
Creating home directory `/home/testuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y_
```

설정의 변경을 저장해주기 위해  
“service vsftpd restart”  
명령어를 통해 서비스를 재시작해줍니다.

이후 “adduser 유저” 명령어를 통해  
자신이 FTP로 접속할 계정을 만들어주면  
FTP 서버에 대한 준비는 끝납니다.



# CentOS

```
[root@localhost ~]# yum install vsftpd ftp
```

FTP의 기능을 사용하기 위해  
ftp 데몬과 ftp 자체를  
"yum install vsftpd ftp"  
명령어를 입력해 설치해줍니다.

```
[root@localhost ~]# service vsftpd status
Redirecting to /bin/systemctl status vsftpd.service
# vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[root@localhost ~]#
```

해당 서비스가 정상적으로 설치되었는지  
확인하기 위해  
"service vsftpd status"  
명령어를 입력해 확인해줍니다.

```
[root@localhost ~]# vi /home/vsftpd.conf
```

해당 conf를 설정해주기 위해  
"vi /home/vsftpd.conf" 으로  
새로운 conf 파일을 만들어줍니다.

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_file=/var/log/xferlog
xferlog_std_format=YES
chroot_local_user=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

```
-- INSERT --
```

왼쪽 사진과 같이 해당 파일에 설정 내용을  
적어줍니다.

설정 내용을 모두 적었다면  
ESC -> :wq 를 입력하고 설정을 종료합니다.

```
[root@localhost ~]# mv /home/vsftpd.conf /etc/vsftpd/vsftpd.conf  
mv: overwrite '/etc/vsftpd/vsftpd.conf'? y  
[root@localhost ~]# _
```

파일 자체의 설정을 적용하기 위해  
“mv /home/vsftpd.conf /etc/vsftpd/vsftpd.conf”  
명령어를 사용하여 파일을 덮어 씌워줍니다.

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=ftp
success
[root@localhost ~]# service vsftpd start
Redirecting to /bin/systemctl start vsftpd.service
[root@localhost ~]# _
```

해당 서비스를 사용하기 위해  
포트 개방이 필요합니다.

"firewall-cmd --permanent --zone=public  
--add-service=ftp"  
명령어를 이용하여 포트를 개방해준 후

"service vsftpd start"  
명령어를 사용하여 서비스를 시작하면  
설정은 마무리됩니다.

# APM 설치



# Ubuntu

```
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

42 packages can be updated.
13 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@user:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 42 not upgraded.
Need to get 1,730 kB of archives.
After this operation, 6,978 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

APM 중 apache를 설치하기 위해  
“apt-get install apache2”  
명령어를 입력 후 Enter를 눌러줍니다.

그 후 설치에 동의하는 문구가 나오면 y를  
누른 후 Enter를 눌러줍니다.

# APM ( Apache 설치 ) [Ubuntu]

```
Setting up apache2-utils (2.4.29-1ubuntu4.1) ...  
Setting up apache2-bin (2.4.29-1ubuntu4.1) ...  
Setting up apache2 (2.4.29-1ubuntu4.1) ...  
Enabling module mpm_event.  
Enabling module authz_core.  
Enabling module authz_host.  
Enabling module authn_core.  
Enabling module auth_basic.  
Enabling module access_compat.  
Enabling module authn_file.  
Enabling module authz_user.  
Enabling module alias.  
Enabling module dir.  
Enabling module autoindex.  
Enabling module env.  
Enabling module mime.  
Enabling module negotiation.  
Enabling module setenvif.  
Enabling module filter.  
Enabling module deflate.  
Enabling module status.  
Enabling module reqtimeout.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
Processing triggers for ureadahead (0.100.0-20) ...  
Processing triggers for systemd (237-3ubuntu10) ...  
Processing triggers for ufw (0.35-5) ...  
root@user:~#
```

해당 창과 같이 정상적으로 설치되었다면 apache2 기능이 설치된 것입니다.

```
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for systemd (237-3ubuntu10) ...
Processing triggers for ufw (0.35-5) ...
root@user:~# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7
  mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 42 not upgraded.
Need to get 21.0 MB of archives.
After this operation, 162 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

이번엔 MySQL을 설치합니다.  
“apt-get install mysql-server”  
해당 명령어를 통해 설치합니다.

이전과 같이 설치에 동의하느냐는  
설명이 나오면 y를 입력후 Enter를 눌러줍니다.

# APM ( MySQL 설치 ) [Ubuntu]

```
Preparing to unpack .../13-libhttp-message-perl_6.14-1_all.deb ...
Unpacking libhttp-message-perl (6.14-1) ...
Selecting previously unselected package mysql-server.
Preparing to unpack .../14-mysql-server_5.7.22-0ubuntu18.04.1_all.deb ...
Unpacking mysql-server (5.7.22-0ubuntu18.04.1) ...
Setting up libhtml-tagset-perl (3.20-3) ...
Setting up libevent-core-2.1-6:amd64 (2.1.8-stable-4build1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Setting up libencode-locale-perl (1.05-1) ...
Setting up libtimedate-perl (2.3000-2) ...
Setting up libio-html-perl (1.001-1) ...
Setting up liblwp-mediatypes-perl (6.02-1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Setting up libaio1:amd64 (0.3.110-5) ...
Setting up liburi-perl (1.73-1) ...
Processing triggers for systemd (237-3ubuntu10) ...
Setting up libhtml-parser-perl (3.72-3build1) ...
Setting up libcgil-pm-perl (4.38-1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up mysql-client-core-5.7 (5.7.22-0ubuntu18.04.1) ...
Setting up libfcgi-perl (0.78-2build1) ...
Setting up libhttp-date-perl (6.02-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mysql-server-core-5.7 (5.7.22-0ubuntu18.04.1) ...
Setting up libcgil-fast-perl (1:2.13-1) ...
Setting up libhttp-message-perl (6.14-1) ...
Setting up mysql-client-5.7 (5.7.22-0ubuntu18.04.1) ...
Setting up mysql-server-5.7 (5.7.22-0ubuntu18.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and mysam-recover options (if present)
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up mysql-server (5.7.22-0ubuntu18.04.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for systemd (237-3ubuntu10) ...
root@user:~#
```

Apache와 같이 정상적으로 설치가 되었는지 확인해줍니다.

```
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Thu Jun  7 02:10:35 UTC 2018

System load:  0.17           Processes:            92
Usage of /:   29.9% of 14.70GB Users logged in:       0
Memory usage: 28%           IP address for enp0s3: 10.0.2.15
Swap usage:   0%

42 packages can be updated.
13 updates are security updates.

root@user:~# sudo apt-get install php libapache2-mod-php php-xml php-gd php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libapache2-mod-php7.2 libfontconfig1 libgd3 libjpeg8
  libjpeg-turbo8 libjpeg8 libtiff5 libwebp6 libxpm4 php php7.2 php7.2-cli php7.2-common php7.2-gd
  php7.2-json php7.2-mysql php7.2-opcache php7.2-readline php7.2-xml
Suggested packages:
  php-pear libgd-tools
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libapache2-mod-php libapache2-mod-php7.2 libfontconfig1
  libgd3 libjpeg8 libjpeg-turbo8 libjpeg8 libtiff5 libwebp6 libxpm4 php php-gd php-mysql php-xml
  php7.2 php7.2-cli php7.2-common php7.2-gd php7.2-json php7.2-mysql php7.2-opcache
  php7.2-readline php7.2-xml
0 upgraded, 25 newly installed, 0 to remove and 42 not upgraded.
Need to get 2,122 kB/5,965 kB of archives.
After this operation, 24.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

PHP는 앞에 두개에 비해 설치할 것이 많습니다.  
“apt-get install php libapache2-mod-php  
php-xml php-gd php-mysql”  
해당 명령어를 통해 설치합니다.

이전과 같이 설치에 동의하느냐는  
설명이 나오면 y를 입력후 Enter를 눌러줍니다.

```
Unpacking php-xml (1:7.2+60ubuntu1) ...
Setting up php7.2-mysql (7.2.5-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/7.2/mods-available/mysqli.ini with new version
Creating config file /etc/php/7.2/mods-available/pdo_mysql.ini with new version
Setting up php7.2-gd (7.2.5-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/gd.ini with new version
Setting up php7.2-xml (7.2.5-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/dom.ini with new version
Creating config file /etc/php/7.2/mods-available/simplexml.ini with new version
Creating config file /etc/php/7.2/mods-available/xdiff.ini with new version
Creating config file /etc/php/7.2/mods-available/xml.ini with new version
Creating config file /etc/php/7.2/mods-available/xmlreader.ini with new version
Creating config file /etc/php/7.2/mods-available/xmlwriter.ini with new version

Creating config file /etc/php/7.2/mods-available/xsl.ini with new version
Setting up php-xml (1:7.2+60ubuntu1) ...
Setting up php-gd (1:7.2+60ubuntu1) ...
Setting up libapache2-mod-php7.2 (7.2.5-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/apache2/php.ini with new version
apache2_invoke: Enable module php7.2
Setting up php-mysql (1:7.2+60ubuntu1) ...
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.5-0ubuntu0.18.04.1) ...
Setting up php (1:7.2+60ubuntu1) ...
root@user:~# _
```

앞에 두개와 같이  
정상적으로 설치되었는지 확인합니다.

# APM 작동확인 [Ubuntu]

```
Ubuntu 18.04 LTS user tty1
user login: root
Password:
Last login: Thu Jun  7 02:16:09 UTC 2018 on tty1
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

42 packages can be updated.
13 updates are security updates.

root@user:~# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@user:~# /etc/init.d/mysql restart
[ ok ] Restarting mysql (via systemctl): mysql.service.
root@user:~#
```

설치한 프로그램들이 잘 작동하는지  
확인하기 위해 Apache와 MySQL을  
“/etc/init.d/기능명 restart”로 재시동해줍니다.



```
root@user:~# netstat -nlp | grep 80
tcp6      0      0 :::80          :::*           LISTEN     1086/apache2
root@user:~# netstat -nlp | grep 3306
tcp      0  0 127.0.0.1:3306  0.0.0.0:*      LISTEN     1152/mysqld
root@user:~#
```

해당 기능의 포트가 정상적으로 작동하는지  
확인하기 위해  
“netstat -nlp” 명령어를 이용해 80포트와  
3306포트가 정상적으로  
사용되고 있는지 확인합니다.

내용 확인이 불가할 경우  
“netstat -nlp | grep 포트” 명령어로  
확인합니다.

정상 작동한다면 설치가 완료된 것입니다.

# CentOS

```
[root@localhost ~]# yum update_
```

yum의 업데이트는 클라우드 플랫폼 특성상  
진행하지 않아야 합니다.

"yum update"

왼쪽과 같은 명령어를 수행후 커널 버전이  
변경되면, OS와 클라우드 플랫폼간에 차이가  
발생되면서 OS가 정상적으로 동작이 불가능할 수  
있습니다.

yum update 명령어를 수행하였을 경우 이후에  
문제가 발생하는 사항에 대하여서는 지원이  
불가능합니다.

```
[root@localhost ~]# yum install httpd mariadb mariadb-server php php-mysql
```

본장에서 OS는 CentOS 7기준입니다.  
(버전에 따라 차이가 있습니다.)

APM을 설치를 진행합니다.

“yum install httpd mariadb mariadb-server php php-mysql”

위 명령어로 설치합니다.

MySQL이나 같은 기반으로 만들어져 거의 같은  
기능을 수행하는 mariaDB를 설치해줍니다.  
(본 안내는 mariaDB를 기준으로 설치하였습니다.)

# APM 설치 [CentOS]

```
===== Size
Installing:
httpd                x86_64      2.4.6-80.el7.centos    base      2.7 M
mariadb              x86_64      1:5.5.56-2.el7        base      8.7 M
mariadb-server       x86_64      1:5.5.56-2.el7        base      11 M
php                  x86_64      5.4.16-45.el7         base      1.4 M
php-mysql            x86_64      5.4.16-45.el7         base      101 k
Installing for dependencies:
perl-Compress-Raw-Bzip2  x86_64      2.061-3.el7           base      32 k
perl-Compress-Raw-Zlib  x86_64      1:2.061-4.el7         base      57 k
perl-DBD-MySQL          x86_64      4.023-6.el7           base      140 k
perl-DBI                x86_64      1.627-4.el7           base      802 k
perl-Data-Dumper        x86_64      2.145-3.el7           base      47 k
perl-IO-Compress         noarch      2.061-2.el7           base      260 k
perl-Net-Daemon          noarch      0.48-5.el7            base      51 k
perl-PIRPC              noarch      0.2020-14.el7         base      36 k

Transaction Summary
=====
Install 5 Packages (+8 Dependent packages)

Total download size: 25 M
Installed size: 124 M
Is this ok [y/d/N]: y
```

해당 프로그램들을 설치하는 것에 대한  
동의를 하기 위해 y를 입력한 뒤  
Enter를 눌러줍니다.

# APM 설치 [CentOS]

```
Verifying : perl-PIRPC-0.2020-14.el7.noarch 7/13
Verifying : php-mysql-5.4.16-45.el7.x86_64 8/13
Verifying : 1:perl-Compress-Raw-Zlib-2.061-4.el7.x86_64 9/13
Verifying : 1:mariadb-server-5.5.56-2.el7.x86_64 10/13
Verifying : perl-IO-Compress-2.061-2.el7.noarch 11/13
Verifying : perl-DBI-1.627-4.el7.x86_64 12/13
Verifying : 1:mariadb-5.5.56-2.el7.x86_64 13/13
```

```
Installed:
  httpd.x86_64 0:2.4.6-80.el7.centos      mariadb.x86_64 1:5.5.56-2.el7
  mariadb-server.x86_64 1:5.5.56-2.el7    php.x86_64 0:5.4.16-45.el7
  php-mysql.x86_64 0:5.4.16-45.el7
```

```
Dependency Installed:
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7
  perl-DBD-MySQL.x86_64 0:4.023-6.el7
  perl-DBI.x86_64 0:1.627-4.el7
  perl-Data-Dumper.x86_64 0:2.145-3.el7
  perl-IO-Compress.noarch 0:2.061-2.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7
  perl-PIRPC.noarch 0:0.2020-14.el7
```

```
Complete!
[root@localhost ~]# _
```

설치가 완료되었는지 확인합니다.

```
[root@localhost ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
# httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[root@localhost ~]# service mariadb status
Redirecting to /bin/systemctl status mariadb.service
# mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor pre
   Active: inactive (dead)
[root@localhost ~]# php -v
PHP 5.4.16 (cli) (built: Apr 12 2018 19:02:01)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
[root@localhost ~]# _
```

기능이 정상적으로 동작하는지 확인하기 위해  
“service 기능명 status”  
명령어를 통해 확인합니다.

PHP의 경우는 php -v 명령어를 통해  
설치가 되었는지 확인합니다.

# APM 기능 시작 [CentOS]

```
[root@localhost ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@localhost ~]# service mariadb start
Redirecting to /bin/systemctl start mariadb.service
[root@localhost ~]# _
```

방금 확인된 기능들이 정상적으로  
설치된 것으로 보이면 해당 기능을  
“service 기능명 start” 로 시작해줍니다.



```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=http
success
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=mysql
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# _
```

정상적인 서비스를 하기 위해  
해당 기능들을 방화벽에 등록해줍니다.  
"firewall-cmd --permanent --zone=public --add-  
service=기능명"  
위 명령어를 통해 추가해준 후  
"firewall-cmd --reload"  
명령어를 통해 리로드해줍니다.

설정을 마치면 정상적으로 기능 이용이 가능합니다.

(firewall-cmd는 CentOS7에 내포된 firewalld에 대한 적용 명령어입니다.  
익숙치 않은 경우 해당 firewalld를 삭제하고, 웹콘솔 방화벽으로 접근제어를  
진행하거나 서버내 iptables를 활용하시기 바랍니다.)