

# Hoing

Database, Oracle, MySQL, Linux, Cloud, NoSQL and Development



## CentOS 8 APM(Apache + PHP + MariaDB) 설치 및 연동 구성

작성자 [태랑\(정현호\)](#) 1월 3, 2021

Last Updated on 7월 21, 2021 by [태랑\(정현호\)](#)

### APM은 Apache PHP MariaDB 의 약자

APM 은 웹 서버를 구축하고 PHP 로 작성된 웹 어플리케이션을 구동하기 위해 이전부터 많이 사용해 온 스택 입니다.

PHP 를 구동 하기 위한 서버 스택의 구성은 여러가지 방식으로 할 수 있습니다.

Apache + PHP(mod\_php) + Mysql 나 MariaDB

Apache + PHP-FPM + Mysql 나 MariaDB

Nginx + PHP-FPM + Mysql 나 MariaDB

이번 글에서는 dnf(rpm)으로 Apache + PHP-FPM + MariaDB 기술 하도록 하겠습니다.

### 설치 환경

OS: CentOS 8.3

Apache 2.4.37 + PHP-FPM 7.4.13 + MariaDB 10.3.17

설치순서는 Apache MariaDB PHP-FPM 순으로 설치 하겠습니다

[참고] CentOS 7 버전에서의 APM 설치 구성은 아래 포스팅을 참조하시면 됩니다.



**리눅스에 APM(Apache + PHP + MySQL) 설치 및 연동**

APM은 Apache PHP MariaDB 의 약자 APM 은 웹 서버를 구축하고 PHP 로 작성된 웹 어플리케이션을 구동하기 위해 이전부터 많이 사용해온 스택 입니다.  
CentOS 7.8 버전에서의 구성 내용을 포함하고 있습니다.

## 0. 사전 환경 구성

외부 Repository 구성

```
user$ sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
user$ sudo dnf -y install https://rpms.remirepo.net/enterprise/remi-release-8.rpm
```

사전 필요 패키지 설치

```
user$ sudo dnf -y install gcc gcc-c++ pcre-devel \
make git openssl wget vim unzip GeolIP-devel \
```

```
git tree cmake expat-devel libxml2-devel ₩
openssl openssl-devel python3 curl gnupg2 ₩
libxslt-devel gd-devel perl-ExtUtils-Embed
```

## 1. Apache 설치

### Apache 설치

```
user$ sudo dnf -y install httpd httpd-devel apr ₩
apr-util apr-devel apr-util-devel apr-util-openssl ₩
mod_proxy_html apr-util libnghttp2 mod_http2 ₩
mod_ssl httpd-tools
```

### 버전 확인

```
user$ curl --head http://localhost/
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1g
```

```
user$ httpd -V
Server version: Apache/2.4.37 (centos)
Server built: Nov 4 2020 03:20:37
```

## 2. MariaDB 설치

### MariaDB 설치

```
user$ sudo dnf -y install mariadb mariadb-common \W
mariadb-devel mariadb-server mariadb-server-utils \W
mariadb-gssapi-server
```

## 기본 파라미터 추가

```
user$ cd /etc/my.cnf.d
user$ sudo vi mariadb-server.cnf
```

[mysqld] <-- 절에 아래 내용을 추가로 입력해 줍니다.

```
user=mysql
server-id=1
log_bin = binlog
expire_logs_days=10
innodb_buffer_pool_size = 384M
innodb_file_per_table=TRUE
# bind-address=0.0.0.0
character-set-client-handshake=OFF
skip-character-set-client-handshake
max_allowed_packet=500M
init_connect=SET collation_connection = utf8mb4_general_ci
init_connect=SET NAMES utf8mb4
character-set-server = utf8mb4
collation-server = utf8mb4_general_ci
```

## MariaDB 시작/중지

MariaDB 중지

```
user$ sudo systemctl stop mariadb
```

MariaDB 기동

```
user$ sudo systemctl start mariadb
```

## 접속 확인

```
user$ sudo mysql
```

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 9

Server version: 10.3.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
MariaDB [(none)]>
```

## 3. PHP 설치

### REMI PHP 7.4 설치

```
user$ sudo dnf -y install php74 php74-php-common \
php74-php-fpm php74-php-gd \
php74 php74-php-devel php74-libzip \
php74-php-mysqlnd php74-php-openssl \
php74-php-mbstring php74-php-pdo \
php74-php-pecl-imagick-devel php74-php-pecl-mcrypt \
php74-php-pecl-zip php74-php-bcmath \
curl curl-devel php74-php-pecl-imagick \
libpng libpng-devel libwebp libwebp-devel \
libXpm libXpm-devel openssl openssl-devel \
autoconf zlib zlib-devel freetype freetype-devel \
gd gd-devel libmcrypt libmcrypt-devel \
libtool-ltdl-devel libxml2-devel \
libxml2 libcurl libcurl-devel php74-php-soap \
php74-php-pecl-crypto php74-php-pecl-mysql
```

## 버전 확인

```
user$ php74 -v
PHP 7.4.13 (cli) (built: Nov 24 2020 10:03:34) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.13, Copyright (c), by Zend Technologies
```

## php.ini 파일 수정

```
user$ cd /etc/opt/remi/php74/
user$ sudo vi php.ini
```

아래 5개는 주석 제거 및 내용 수정

```
date.timezone = Asia/Seoul
expose_php = Off
post_max_size = 500M
upload_max_filesize = 200M
memory_limit = 512M
```

## 4. PHP-FPM 설정

php-fpm.conf 파일 수정

```
user$ cd /etc/opt/remi/php74/  
user$ sudo vi php-fpm.conf
```

```
include=/etc/opt/remi/php74/php-fpm.d/*.conf  
--> 주석이 해제 되어 있는지 확인
```

```
daemonize = yes  
-> 주석으로 되어 있다면 주석을 해제 합니다.
```

```
; Log level  
; Possible Values: alert, error, warning, notice, debug  
; Default Value: notice  
;log_level = notice  
-> PHP-FPM 데몬의 Log level Default 는 notice 입니다.  
-> 문제에 대한 분석이나 초기 설정 후 확인 등이 필요하다면 debug 레벨로 변경하여 사용해도 됩니다.
```

## www.conf 파일 수정

```
user$ cd /etc/opt/remi/php74/php-fpm.d  
user$ sudo cp www.conf www.conf.ori
```

```
user$ sudo vi www.conf
```

```
; pool name ('www' here)  
[www]  
-> PHP Pool 이름을 지정하는 항목 입니다.  
-> 프로그램명이나 사이트 명이나, 업무가 있다면 업무명 등을 입력 하여도 됩니다.  
-> Pool 혹은 Container 의 식별, 이름 입니다 기본으로 www 지정되어 있습니다.  
-> Pool은 conf 파일 별로 여러개의 Pool을 만들 수 있습니다. 이때는 Pool 별로 이름을 지정하여 겹치지 않게 해야 합니다.
```

# 아래 내용과 같이 되어 있는지 확인 및 주석이 되어 있다면 주석을 해제 합니다.

```
user = apache
group = apache
listen.acl_users = apache
```

```
listen = /var/opt/remi/php74/run/php-fpm/www.sock
```

=> 이부분은 접속을 받는 방식을 설정하는 부분으로 Socket(UDS) 방식으로 사용 하거나 IP 방식으로 사용하거나 둘 중 하나를 선택하여 설정해야 합니다.

=> IP 방식으로는 아래와 같이 지정 할 수 있습니다.

```
listen = 123.123.123.123:9000
```

```
listen = 127.0.0.1:9000
```

```
listen = 9000 <== 이와 같이 IP 없이 포트만 기재하면 모든 주소에서 접속이 가능한 상태 입니다
(listen on a TCP socket to all addresses)
```

소켓 방식은 아래와 같이 파일명을 지정해주면 됩니다.

```
listen = listen = /var/opt/remi/php74/run/php-fpm/www.sock
```

포스팅에서는 같은 서버에서 apache 와 php-fpm 사용 하는 설정으로 할 것임으로 socket 을 사용하여 설정하도록 하겠습니다

같은 서버내 라도 IP를 사용 할 수 있고 사용하여도 됩니다.

## PHP-FPM 프로세스

프로세스 수와 관련된 pm 으로 시작하는 파라미터 가 있습니다

```
pm = dynamic
pm.max_children = 50
pm.start_servers = 5
```

```
pm.min_spare_servers = 5
pm.max_spare_servers = 40
pm.max_requests = 500
```

\* 위의 설정값은 예시입니다.

pm.max\_children 는 설정된 PHP Pool(여기서는 www) 내에서 가동할 수 있는 최대 자식 프로세스 수를 의미 합니다

pm.start\_servers 는 php-fpm 을 실행할 때 초기에 생성하는 자식 프로세스의 개수입니다

pm.min\_spare\_servers 는 idle 상태의 자식 프로세스 개수가 이 개수보다 작으면 자식 프로세스를 생성 합니다

pm.max\_spare\_servers 는 idle 상태의 자식 프로세스의 최대 개수를 의미 합니다.

pm.max\_requests 는 각 프로세스가 최대 request를 처리하면 수 입니다.  
500 으로 설정되어 있다면 500번 request 후 프로세스를 다시 생성 합니다  
프로세스가 점진적인 메모리 증가에 대한부분을 방지 할 수 있습니다.  
apache 에서 유사한 설정으로는 MaxRequestsPerChild 있습니다.

## [참고] PHP-FPM 메모리 사용현황 및 세션 계산

아래 명령어를 이용하면 현재 fork된 php-fpm 프로세스 당 메모리 사용량을 확인 할 수 있습니다.

```
user$ ps --no-headers -o "rss,cmd" -C php-fpm | awk '{ sum+=$1 } END { printf ("%d%s\n", sum/NR/1024,"M") }'
```

시스템에 가용한 메모리나 사용 할 정도에서 프로세스당 사용메모리를 나누어서 최대 자식프로세스 개수를 산정 해도 되겠습니다

ex) PHP에 6GB 를 사용 하려고 함 , PHP-FPM이 현재 사용중인 메모리량이 대략 60MB 임

6000MB / 60MB = 자식 프로세스 개수 100개 로 설정

이와 같이 각 시스템 마다 메모리 상황을 고려하여 적절한 pm.max\_children 값을 선정하여 설정하면 되  
겠습니다.

## 로그 설정

로그 관련된 내용의 주석 해제와 변경을 합니다

```
access.log = /var/opt/remi/php74/log/php-fpm/$pool.access.log
```

-> 주석 해제 및 경로 지정

```
access.format = "%R - %u %t W"%m %r%Q%qW" %s %f %{mili}d %{kilo}M %C%%"
```

--> 주석제거

```
slowlog = /var/opt/remi/php74/log/php-fpm/$pool.slow.log
```

--> 주석제거, 파일명 변경

```
request_slowlog_timeout = 5
```

--> 주석제거, 내용변경

--> 단위는 초로 5는 예시 입니다. 해당 시간은 선택해서 변경 하시면 됩니다.

```
catch_workers_output = yes
```

-> 주석 제거

-> main error log 파일에 worker 의 에러 로그가 같이 기록 되게 됩니다.

```
php_admin_value[error_log] = /var/opt/remi/php74/log/php-fpm/www-error.log
```

```
php_admin_flag[log_errors] = on
```

-> 주석 제거

## PHP Opcache 파라미터

[참고] PHP 5.5 버전 부터 기본에 포함되었으며 이전에는 zend optimizer 를 별도로 설치 및 plugin 하여 사용 하였습니다.

사용 용도는 컴파일 / 실행 / 결과 의 순으로 처리되는 PHP에 대해서 Opcache(예전의 Zend Optimizer) 를 이용하여 컴파일된 PHP 코드를 메모리에 Cache 해서 접속요청에 더 빠르게 응답을 줄수 있는 기능 입니다

php7.x 버전부터 Opcache 의 기본 메모리가 128M 입니다.  
메모리를 변경하거나 다른 파라미터를 조정 하고자 한다면 10-opcache.ini 에서 설정할 수 있습니다.

```
user$ cd /etc/opt/remi/php74/php.d  
user$ sudo vi 10-opcache.ini
```

10-opcache.ini 파일에는 opcache 의 활성화 내용부터 해서 기본적으로 사용 될수 있는 파라미터는 주석이 해제되어 활성화가 되어 있는 상태 입니다.

opcache의 파라미터의 간략한 설명은 아래와 같습니다.

opcache.enable=1 - Zend Opacache 활성화 여부 결정 , 1일때 활성화 됨

opcache.enable\_cli=1 - PHP CLI 버전에서(명령행 버전에서도) Zend Opacache 활성화 여부 결정

opcache.memory\_consumption - 캐쉬에 사용할 메모리 크기. 7.0부터 기본 크기는 128M

opcache.interned\_strings\_buffer=8 - 메모리에서 interned 문자열을 위해 사용할 메모리량

opcache.fast\_shutdown=1 - 활성화 시 accelerated code 에 대해서 사용되어짐

opcache.interned\_strings\_buffer - Opcache 가 사용하는 내부 스트링( class name 이나 file등)을 저장하기 위한 메모리의 크기.

opcache.max\_accelerated\_files - Opcache 가 캐쉬할 최대 파일 갯수.

opcache.validate\_timestamps - Default 1 / 0(off)이 아닐 경우 opcache.revalidate\_freq 에 설정된 시간마다 코드와 캐쉬의 변경 여부를 체크한다.

validate\_timestamps 가 꺼져 있을 경우 코드의 변경 여부를 체크하지 않으므로 PHP 소스가 수정되어도 반영되지 않는다.

그러므로 변경 사항을 반영하려면 httpd 를 재기동하거나 httpd 에 SIGUSR2 시그널을 전송해야 한다.

opcache.file\_cache\_only=1 - makes sure that the in-memory opcache is disabled and only files are used, which is what you want for CLI.

1 로 설정할 경우 in-memory opcache 사용이 안됨 = php-fpm 에서 사용할수 없음

그래서 메모리 관련하여 수정이 필요 할 경우 아래와 같이 용량을 늘려서 사용 할 수 있습니다.

opcache.memory\_consumption=256

-> 입력의 예시 이며 단위는 MB 입니다

-> 이부분(메모리 변경) 은 필수는 아닙니다.

-> 사용시 앞에 주석(;) 을 제거 합니다.

## php-fpm 재시작

```
[root]]# sudo systemctl restart php74-php-fpm
```

\* PHP-FPM 설치 후 자동으로 기동되어 있을수 있음

## 프로세스 및 포트 오픈 확인

```
[root]]# ps -ef| grep php
root 62849 1 0 11:26 00:00:00 php-fpm: master process (/etc/php-fpm.conf)
apache 62850 62849 0 11:26 00:00:00 php-fpm: pool www
apache 62851 62849 0 11:26 00:00:00 php-fpm: pool www
apache 62852 62849 0 11:26 00:00:00 php-fpm: pool www
apache 62853 62849 0 11:26 00:00:00 php-fpm: pool www
apache 62854 62849 0 11:26 00:00:00 php-fpm: pool www
```

## listen을 Unix Socket 으로 사용 할 경우

```
[root]]# ls -al /var/opt/remi/php74/run/php-fpm/
srw-rw----+ 1 root root 0 11월 29 11:35 www.sock
```

## listen을 IP 으로 사용 할 경우

```
[root]]# netstat -antp | grep 9000
tcp 0 0 127.0.0.1:9000 0.0.0.0:* LISTEN 6389/php-fpm: master
```

## apache 설정을 변경

/etc/httpd/conf/httpd.conf 파일 수정

```
user$ sudo vi /etc/httpd/conf/httpd.conf
```

```
ServerName www.example.com:80
```

-> 주석 해제

-> 아래 쪽에 2개 내용추가

```
ServerTokens Prod
```

```
ServerSignature Off
```

아래 Directory 사이의 내용을 변경 합니다.

```
<Directory "/var/www/html">
```

.... 사이의 내용...

```
</Directory>
```

```
#Options Indexes FollowSymLinks
```

-> 주석 처리 후 아래 내용으로 입력

```
Options MultiViews FollowSymLinks
```

-> 내용 추가

```
#AllowOverride None
```

-> 주석 처리 후 아래 내용으로 입력

```
AllowOverride All
```

-> 내용 추가

## 00-proxy.conf 파일 수정

```
user$ cd /etc/httpd/conf.modules.d/
```

```
user$ sudo vi 00-proxy.conf
```

아래 4개 내용 추가

```
ProxyRequests Off
ProxyErrorOverride on
ProxyTimeout 600
ProxyPreserveHost On
```

[참고] PHP-FPM 과 연결되는 Proxy\_Pass 설정은 아래 파일내에 대부분이 설정되어 있습니다.  
/etc/httpd/conf.d/php74-php.conf

## 6. 방화벽 및 Apache 재시작

OS 방화벽을 오픈 합니다.

```
# 80 포트 추가
user$ sudo firewall-cmd --permanent --add-port=80/tcp
```

```
# 80 포트 삭제(참고)
user$ sudo firewall-cmd --permanent --remove-port=80/tcp
```

```
# 정책 재반영 및 방화벽 정책 확인
user$ sudo firewall-cmd --reload;firewall-cmd --list-all
```

PHP 정상 작동 여부 확인을 위해 phpinfo.php 파일을 생성 합니다.

```
[root]]# cd /var/www/html  
[root]]# sudo vi phpinfo.php
```

```
<?php  
phpinfo();  
?>
```

apache 재시작

```
user$ sudo systemctl restart httpd
```

접속 테스트

```
http://ip주소/phpinfo.php
```

---

## PHP Version 7.4.13

System	Linux wmp 4.18.0-240.el8.x86_64 #1 SMP Fri Sep 25 19:48:47 UTC 2020 x86_64
Build Date	Nov 24 2020 10:03:34
Build System	Red Hat Enterprise Linux release 8.3 (Ootpa)
Build Provider	Remi's RPM repository <https://rpms.remirepo.net/>
Server API	FPM/FastCGI ←
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/opt/remi/php74
Loaded Configuration File	/etc/opt/remi/php74/php.ini
Scan this dir for additional .ini files	/etc/opt/remi/php74/php.d
Additional .ini files parsed	/etc/opt/remi/php74/php.d/10-opcache.ini, /etc/opt/remi/php74/php.d/20-bcmath.ini, bz2.ini, /etc/opt/remi/php74/php.d/20-calendar.ini, /etc/opt/remi/php74/php.d/20-ctype.ini, /etc/opt/remi/php74/php.d/20-curl.ini, /etc/opt/remi/php74/php.d/20-exif.ini, /etc/opt/remi/php74/php.d/20-ftp.ini, /etc/opt/remi/php74/php.d/20-gd.ini, /etc/opt/remi/php74/php.d/20-iconv.ini, /etc/opt/remi/php74/php.d/20-json.ini, /etc/opt/remi/php74/php.d/20-mbstring.ini, /etc/opt/remi/php74/php.d/20-mysqli.ini, /etc/opt/remi/php74/php.d/20-mysqlnd.ini, /etc/opt/remi/php74/php.d/20-phar.ini, /etc/opt/remi/php74/php.d/20-sockets.ini, /etc/opt/remi/php74/php.d/20-sqlite3.ini, /etc/opt/remi/php74/php.d/20-tokenizer.ini, /etc/opt/remi/php74/php.d/30-pdo_mysql.ini, /etc/opt/remi/php74/php.d/30-mysqli.ini, /etc/opt/remi/php74/php.d/30-pdo_mysql.ini, /etc/opt/remi/php74/php.d/40-imagick.ini, /etc/opt/remi/php74/php.d/40-imagick.ini
PHP API	20190902
PHP Extension	20190902

이어지는 글



### Apache 2.4 에 mod\_cband 와 웹 방화벽 ModSecurity 추가

ModSecurity는 웹 서비스의 공격을 효과적으로 차단할 수 있는 오픈소스 웹 방화벽 (WAF) 로써 OWASP Core Ruleset을 사용할 수 있습니다.



### 웹 기반 mysql 관리툴 - phpmyadmin 설치 및 설정

웹 기반의 mysql 관리툴PHP로 개발 된 오픈소스이고, 웹 기반의 mysql 관리 툴 입니다. 사용하는 php 버전에 따라서 phpmyadmin을 선택하셔서 설치하시면 됩니다.

### Apache - 가상호스트 (VirtualHost)를 통한 여러 사이트 운영

한 서버에서 여러 도메인이나 서브 도메인 별 웹서비스



연관된 다른 글



### 우분투(Ubuntu) APM - Apache PHP Mysql 구성 및 설정

APM은 Apache + PHP + Mysql 의 약자 APM은 웹 서버를 구축하고 PHP 로 작성된 웹 어플리케이션을 구동하기 위해 이전부터 많이 사용해온 스택 입니다.



### CentOS 7.8 APM(Apache + PHP + MySQL) 설치 및 연동

CentOS 7.8 환경에서 Apache PHP-FPM MySQL 연동 설치 포스팅 입니다. 오라클 클라우드 프리티어 연재글의 중간글 입니다.



### CentOS 7.8 Nginx PHP Mysql 설치 및 연동 구성 - LEMP

웹 어플리케이션을 구현하기 위해서 필요한 PHP 스택 LEMP는 Linux + Nginx + PHP-FPM + Mysql(MariaDB) 를 의미하는 약자이며 PHP 어플리케이션 구현을 위한 Stack 입니다

우분투(Ubuntu) 18.04 - LEMP(Nginx PHP MariaDB) 1 - 설치 및 연동 설정



웹 어플리케이션을 구현하기 위해서 필요한 PHP 스택 LEMP는 Linux + Nginx + PHP-FPM + Mysql(MariaDB) 를 의미하는 약자이며 PHP 어플리케이션 구현을 위한 Stack 입니다



### CentOS 지원 정책 변경/지원 종료 및 새로운 RHEL Clone 배포판 소식 - Rocky Linux

CentOS는 리눅스의 배포판 중 하나로 Community Enterprise Operating System 의 약자입니다. RHEL의 소스를 기반으로 만들어지며 RHEL에서 최대한 추가나 제거를 자제하는 것을 원칙으로 가장 유사하게 Rebuild 한 배포판 입니다.



#### 태랑(정현호)

Senior DBA(MySQL, Oracle) - 현재 위메프에서 많은 새로움을 경험중입니다  
At WeMakePrice / Previous - Oracle Korea ACS Support / Fedora Kor UserGroup 운영중  
Database 외에도 NoSQL , Linux , Cloud, Http/PHP CGI 등에도 관심이 있습니다  
purityboy83@gmail.com / admin@hoing.io

hoing.io



이전 글 CentOS Nginx 에 Certbot을 통한 무료 SSL인증서 적용 및 HTTP/2 적용

다음글 CentOS 8 LEMP(Nginx + PHP + MariaDB) 설치 및 연동 구성

카테고리 Linux

태그 linux php 설치, mod\_proxy, proxypass, php 7.4 yum, yum php 7.4, yum php74 설치, yum remi php 7.4, rpm apache, linux apache, 리눅스 apache, 리눅스 아파치 설치, linux 아파치 설치, remi php 7.4, 리눅스 php 설치, yum apache 설치, yum mysql 설치, lamp 설치, LAMP Install, apm 설치 순서, 아파치 PHP-FPM, 제로보드XE, Rhymix, 라이믹스, CentOS 8, php 74, Linux, yum, 아파치, MySQL, PHP, php-fpm, apm, 웹서비스, apache php연동, mod\_php, remi, 리눅스 APM 설치, Linux APM 설치, 리눅스 APM 연동, Linux APM 연동, 리눅스 APM 구축, php-fpm 연동, php 7.4, php-fpm 7.4, php74, 리눅스

답글 남기기

댓글

이름 \*

이메일 \*

웹사이트

다음 번 댓글 작성을 위해 이 브라우저에 이름, 이메일, 그리고 웹사이트를 저장합니다.

Notify me via e-mail if anyone answers my comment.

댓글 달기

검색



카테고리

DATA  
BASE

Platfor  
m

DevOp  
s

빅데이  
터

WordPress(  
워드프레스)

JAVA / C /  
그외 개발언  
어

넓은 세상의  
이야기들

최신 글



## BUG.MYSQL.COM

컨테이너와

Docker (5) - 도커

이미지 생성 -

export - Docker

Compose 7월 19,  
2021

컨테이너와

Docker (4) -

commit -

volume mount -

데이터 저장 7월  
7, 2021

MySQL 8.0 변경

된 routine 조회

권한 -

mysql.proc -

SHOW\_ROUTINE

-

information\_schei

7월 6, 2021

소주 디스펜서 -

SOJU DISPENSER

- 로이체 소주메

이트 디스펜서 -

ROYCHE 7월 3,

2021

컨테이너와

Docker (3) - 도커

네트워크 - OS 컨

▶ #104365:

'charmap' codec

can't decode

byte 0x9d in

position Y:

character 7월 20,

2021

▶ #104364:

Workbench can't  
change

configuration file

7월 20, 2021

▶ #104358: typo

in documents 7

월 20, 2021

▶ #104356:

Semi-sync

replication

switched OFF 7

월 20, 2021

▶ #104351:

CJCommunicator

Cannot read

packet header

with

## 태그

---

apache

aws

aws cloud

aws 클라우드

aws 프리티어

fedora

Fedora Linux

free tier

LAMP Install

lamp 설치

Linux

MariaDB

MySQL

MySQL 5.7

MySQL  
Replication

mysql 복제

nginx

OCI

oracle

oracle cloud

oracle cloud  
wordpress

PHP

php-fpm

Rhymix

Solaris

- ubuntu
- vps
- vps 무료
- wordpress
- 라이믹스
- 리눅스
- 무료 vps
- 무료 호스팅
- 솔라리스
- 아마존 aws 클라우드는
- 아마존 클라우드
- 아파치
- 오라클
- 오라클 클라우드
- 우분투 클라우드
- 워드프레스
- 제로보드XE
- 클라우드 우분투
- 페도라
- 프리티어

## SITE STATISTICS

---

**Visitors today :**

348

**Page views today**

: 435

**Total visitors :**

101,534

Total page view:

126,175

---

#### LINK

---

- ▶ [Blog](#)
- ▶ [About ME](#)
- ▶ [Facebook](#)
- ▶ [LinkedIn](#)
- ▶ [Fedoralinux.or.kr](#)
- ▶ [RSS Feed All](#)

#### **MYSQL SERVER TEAM**

---

- ▶ [MySQL Shell 8.0.24 – What’s New? 4월 21, 2021](#)
- ▶ [InnoDB: Converting old atomic code to C++11 4월 21, 2021](#)
- ▶ [The MySQL 8.0.24 Maintenance Release is Generally Available 4월 20, 2021](#)

#### POWERED BY

---



WORDPRESS

NGINX

