

# db.tips

사이트 관리를 위한 페이지들

- [사이트 소개](#)
- [즈보드](#)
- [my설치](#)
- [gg](#)
- [mysql pdf 받은거](#)

# 사이트 소개

Data와 AI에 대해 공부하며 정리한 것들

DA / TA / AA들이 관심 가질 만한 주제들

이런 주제를 가지고 문서 형태로 정리해 나가는 사이트 입니다.

ス ヒ ㇿ

# my설치

30d89c39-4efc-11eb-9f39-eec956d5f988:255

- MySQL, MariaDB 테스트 서버

[컨테이너 정리하고 CentOS 7 기반으로 재생성]

```
docker stop MySQL
```

```
docker stop MariaDB
```

```
docker rm MySQL
```

```
docker rm MariaDB
```

```
docker create --name="MySQL" -h MySQL -p 3307:3306 --restart unless-stopped -i -t docker.io/centos:7 /bin/bash --init-file /root/start.sh
```

```
docker create --name="MariaDB" -h MariaDB -p 3306:3306 --restart unless-stopped -i -t docker.io/centos:7 /bin/bash --init-file /root/start.sh
```

```
docker start MySQL
```

```
docker start MariaDB
```

[MySQL 설치]

1. 컨테이너 접속

```
docker exec -ti MySQL /bin/bash -l
```

2. OS 패키지 설치

```
yum install net-tools wget rsync openssl098e lsof which perl perl-Data-Dumper libaio* numactl-libs sudo -y
```

```
yum update -y
```

3. 로케일 설정

```
rm -f /etc/localtime
```

```
ln -s /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

```
localedef -v -c -i en_US -f UTF-8 en_US.UTF-8
```

```
localedef -v -c -i ko_KR -f UTF-8 ko_KR.UTF-8
```

4. root 계정 암호 변경 / mysql 계정 생성 및 암호 변경

```
passwd
```

```
useradd mysql -d /mysql
```

```
passwd mysql
```

5. 디렉토리 생성

```
export DBNM=testdb
```

```
mkdir -p /mysql/$DBNM
```

```
mkdir -p /mysql/$DBNM/scripts
```

```
mkdir -p /data/$DBNM/
```

```
mkdir -p /data/$DBNM/datadir
```

```
mkdir -p /data/$DBNM/tmpdir
```

```
mkdir -p /data/$DBNM/logdir
```

```
mkdir -p /data/$DBNM/logdir/oldlog
mkdir -p /data/$DBNM/binlog
mkdir -p /data/$DBNM/relaylog
chown -R mysql:mysql /mysql
chown -R mysql:mysql /data
mkdir -p /backup/$DBNM/binary_backup
mkdir -p /backup/$DBNM/data_backup
mkdir -p /backup/$DBNM/data_backup/backup_log
mkdir -p /backup/$DBNM/data_backup/binlog_backup
chown -R mysql:mysql /backup
```

## 6. MySQL 설치파일 준비

```
cd /mysql/$DBNM

wget https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.31-linux-glibc2.12-x86_64.tar.xz

wget https://downloads.percona.com/downloads/percona-toolkit/3.5.0/binary/tarball/percona-toolkit-3.5.0_x86_64.tar.gz

wget https://downloads.percona.com/downloads/Percona-XtraBackup-LATEST/Percona-XtraBackup-8.0.30-23/binary/tarball/percona-xtrabackup-8.0.30-23-Linux-x86_64.glibc2.17.tar.gz

chown mysql:mysql -R /mysql/$DBNM
```

## 7. mysql 계정으로 변경

```
su - mysql

export DBNM=testdb

cd $DBNM
```

## 8. 설치 파일 압축 풀기

```
tar xvf *.xz

tar xvfz *xtrabackup*.tar.gz

tar xvfz *toolkit*.tar.gz
```

## 9. 설치 파일 삭제

```
rm -f *.xz

rm -f *.tar.gz
```

## 10. 디렉토리 명칭 심볼릭 링크 생성

```
ln -s mysql* instance

ln -s *xtrabackup* xtrabackup

ln -s *toolkit* toolkit
```

## 11. /data/\$DBNM/my.cnf 설정 편집

```
vi /data/$DBNM/my.cnf
```

```
[mysqld]
```

```
port = 3306
```

```
socket = /tmp/mysql.sock
```

```
user = mysql
server-id = 1

basedir = /mysql/testdb/instance
datadir = /data/testdb/datadir
tmpdir = /data/testdb/tmpdir
character-set-server = UTF8MB4
collation-server = UTF8MB4_BIN
default_storage_engine = InnoDB
event-scheduler = 0
sysdate-is-now
performance_schema = on
back_log = 100
max_connections = 500
max_connect_errors = 100
thread_cache_size = 10
table_open_cache = 2048
wait_timeout = 86400
max_allowed_packet = 32M
max_heap_table_size = 128M
tmp_table_size = 128M
sort_buffer_size = 256K
join_buffer_size = 256K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
transaction_isolation = READ-COMMITTED
plugin-load = auth_socket=auth_socket.so
thread_stack = 240K
mysqlx=0

### InnoDB Specific options ###
innodb_data_file_path = ibdata1:100M:autoextend
innodb_file_per_table = 1
innodb_buffer_pool_size = 512M
innodb_log_buffer_size = 16M
innodb_log_file_size = 16M
innodb_log_files_in_group = 4
innodb_open_files = 1024
innodb_autoextend_increment = 256
```

```
innodb_flush_log_at_trx_commit = 1
```

```
innodb_flush_method=O_DIRECT
```

```
innodb_change_buffering='all'
```

```
innodb_lock_wait_timeout = 120
```

```
### MyISAM Specific options ###
```

```
key_buffer_size = 64M
```

```
bulk_insert_buffer_size = 64M
```

```
myisam_sort_buffer_size = 128M
```

```
myisam_max_sort_file_size = 4G
```

```
### Logging options ###
```

```
log_bin = /data/testdb/binlog/binlog
```

```
binlog_expire_logs_seconds = 604800
```

```
general_log = 0
```

```
slow_query_log = 1
```

```
long_query_time = 2
```

```
log_error = /data/testdb/logdir/error.log
```

```
slow_query_log_file = /data/testdb/logdir/slow.log
```

```
general_log_file = /data/testdb/logdir/general.log
```

12. mysql 계정의 .bash\_profile 수정

```
echo ">> ~/.bash_profile"
```

```
echo export DBNM=$DBNM >> ~/.bash_profile
```

```
echo export PATH=/mysql/$DBNM/instance/bin:/mysql/$DBNM/xtbackup/bin:/mysql/$DBNM/toolkit/bin:$PATH >> ~/.bash_profile
```

```
echo alias cdmy="cd /mysql/$DBNM/instance" >> ~/.bash_profile
```

```
echo alias cdsp="cd /mysql/$DBNM/scripts" >> ~/.bash_profile
```

```
echo alias cddt="cd /data/$DBNM/datadir" >> ~/.bash_profile
```

```
echo alias cdlog="cd /data/$DBNM/logdir" >> ~/.bash_profile
```

```
echo alias showerr="vi /data/$DBNM/logdir/error.log" >> ~/.bash_profile
```

```
echo alias tailerr="tail -f /data/$DBNM/logdir/error.log" >> ~/.bash_profile
```

```
source ~/.bash_profile
```

13. 데이터베이스 초기화

```
cdmy
```

```
./bin/mysqld --defaults-file=/data/$DBNM/my.cnf --initialize
```

14. 초기 설정 및 운영 스크립트 추가

#1.데이터베이스 기동 스크립트 작성

```
cat > /mysql/$DBNM/scripts/startup.sh << EOF
```

```
#!/bin/sh
```

```
source ~/mysql/.bash_profile
```

```
cd /mysql/$DBNM/instance  
./bin/mysqld_safe --defaults-file=/data/$DBNM/my.cnf &
```

EOF

## #2. 데이터베이스 중단 스크립트 작성

```
cat > /mysql/$DBNM/scripts/shutdown.sh << EOF
```

```
#!/bin/sh
```

```
source ~mysql/.bash_profile
```

```
cd /mysql/$DBNM/instance
```

```
./bin/mysqladmin shutdown -u root -p
```

EOF

## #3. 데이터베이스 접속 스크립트 작성

```
cat > /mysql/$DBNM/scripts/conn.sh << EOF
```

```
#!/bin/sh
```

```
source ~mysql/.bash_profile
```

```
cd /mysql/$DBNM/instance
```

```
./bin/mysql -h localhost -u root -p
```

EOF

## #4. 로그파일 보관 주기 관리용 스크립트 작성

```
cat > /mysql/$DBNM/scripts/logrotate.sh << EOF
```

```
#!/bin/sh
```

```
source ~mysql/.bash_profile
```

```
cd /mysql/$DBNM/instance
```

```
DATE=`date -d '-1 days' +%Y%m%d`\
```

```
mv /data/$DBNM/logdir/error.log /data/$DBNM/logdir/oldlog/error.\${DATE}.log
```

```
mv /data/$DBNM/logdir/slow.log /data/$DBNM/logdir/oldlog/slow.\${DATE}.log
```

```
./bin/mysql -h localhost -u mysql -e "flush logs"
```

```
find /data/$DBNM/logdir/oldlog -name '*.log' -ctime +180 -exec rm -f {} \;
```

EOF

## 15. 데이터베이스 기동

```
cdsp
```

```
sh startup.sh
```

```
ps -ef | grep mysqld
```

```
tailerr
```

## 16. root계정 암호 초기화 // socket 인증 계정 추가

```
export PASS=`cat /data/$DBNM/logdir/error.log | grep tempo | grep pass | awk {'print $NF'}\`
```

```
cdmy
```

```
./bin/mysql -h localhost -u root -p$PASS --connect-expired-password << EOF
```



```
alter user 'root'@'localhost' identified by 'a14231414';
```

```
exit
```

```
EOF
```

```
./bin/mysql -h localhost -u root -pa14231414 << EOF
```

```
create user 'root'@'%' identified by 'a14231414';
```

```
grant all privileges on *.* to 'root'@'%' with grant option;
```

```
grant proxy on "" to 'root'@'%' with grant option;
```

```
create user 'mysql'@'localhost' identified with auth_socket;
```

```
grant all privileges on *.* to 'mysql'@'localhost';
```

```
exit
```

```
EOF
```

## 17. 접속 테스트

- 위에 설정한 계정 정보로 로그인 잘 되는지 확인.

<<OS 계정 mysql에서 DB계정 mysql로 암호 없이 접속>>

```
./bin/mysql -h localhost -u mysql
```

```
exit
```

<<root 계정에 암호 안넣으면 로그인 실패>>

```
./bin/mysql -h localhost -u root
```

<<root 계정에 암호 넣고 접속 확인. 초기 암호는 new1234!>>

```
./bin/mysql -h localhost -u root -p
```

<<DB 접속된 상태에서 계정 정보와 스키마 정보 확인후 DB 접속 종료>>

```
select user,host from mysql.user;
```

```
show databases;
```

```
exit
```

## 18. 자동기동되게 스크립트 수정

```
exit
```

```
vi /root/start.sh
```

```
sudo -u mysql /bin/bash /mysql/testdb/scripts/startup.sh
```

## 19. 히스토리 파일 정리

```
su - mysql
```

```
history -c
```

```
exit
```

```
history -c
```

```
exit
```







# gg

```
#!/bin/bash
# 구동중인 컨테이너로 쉽게 접속하기 위한 스크립트

clear
echo ""
echo " #####"
echo " # 현재 구동 중인 컨테이너  #"
echo " #####"
echo ""

LINE=0

CTNR_NM=`docker ps | grep -v CONTAINER | grep -v CREATED | grep -v STATUS | awk {'print $NF'}`
for i in $CTNR_NM
do
LINE=`expr $LINE + 1`
echo -n " "$LINE" "
echo $i
done

echo "" ; echo ""
echo -n " 접속할 컨테이너: "
read A

if [[ $A -le $LINE ]]
then
B=`expr $A + 0 2> /dev/null`

if [[ $A == $B ]]
then
CTNR=`docker ps | grep -v CONTAINER | grep -v CREATED | grep -v STATUS | awk {'print $NF'} | head -$A 2> /dev/null | tail -1`

echo "" ; echo "" ; echo ""
docker exec -ti $CTNR /bin/bash -l
fi

fi
```

# mysql pdf 받은거

[learning-mysql-get-a-handle-on-your-data-2nbsped-9781492085928.pdf\\_safe.pdf](#)

[Daniel Nichter - Efficient MySQL Performance\\_ Best Practices and Techniques-O'Reilly Media \(2021\).pdf\\_safe.pdf](#)