

Mysql Backup

Index

1. Backup의 종류

2. 실습 1 : Hot Logical Backup

3. 실습 2 : Cold Physical Backup

1. Backup의 종류

Hot Backup

데이터 베이스를 중지하지 않은 상태로 데이터 백업

- 백업하는 동안 서비스가 실행
- 백업하는 동안 데이터가 변경되어 완전한 백업이 안될수 있음

Cold Backup

데이터 베이스를 중지한 상태로 데이터 백업

- 안정적으로 백업이 가능
- 백업하는 동안 서비스가 중단되어야 함

Logical Backup

SQL 문으로 백업

- 느린 속도의 백업과 복원
- 디스크 용량을 적게 사용
- 작업시 시스템 자원을 많이 사용
- 문제 발생에 대한 파악이 쉬움
- 서버 OS 호환이 잘됨

Physical Backup

파일 차체를 백업

- 빠른 속도의 백업과 복원
- 디스크 용량 많이 사용
- 작업시 시스템 자원을 적게 사용
- 문제 발생에 대한 파악과 검토가 어려움
- 서버 OS 호환이 잘안될수 있음

2. 실습 1 : Hot Logical Backup

Backup : 백업

test 데이터 베이스를 test_backup.sql 파일로 백업

mysqldump -u root -p(패스워드) (데이터베이스) > (저장할 파일 이름)

\$ mysqldump -u root -prada test > test_backup.sql

backup shell script 작성 (backup.sh)

backup 파일을 저장할 스크립트 작성

\$ mkdir backup

\$ vi backup.sh

#!/bin/bash

BD=`date +%Y%m%d_%H%M --date=today`

FILE=\${BD}.sql

cd backup

echo "mysqldump -u root -prada test > \$FILE"

mysqldump -u root -prada test > \$FILE

backup.sh 스크립트 실행

\$ /bin/bash backup.sh

crontab을 이용하여 주기적으로 명령을 실행 : 5분에 한번씩 백업

```
$ crontab -e
```

```
# /usr/bin : mysqldump를 실행
```

```
# /bin : /bin/bash를 실행, backup.sh에 있는 date 명령어가 포함
```

```
PATH=/usr/bin:/bin:/home/ubuntu/.pyenv/versions/python3/bin
```

```
*/5 * * * * /bin/bash /home/ubuntu/backup.sh
```

Restore : 복원

```
# 데이터 베이스 생성
```

```
$ mysql -u root -p
```

```
mysql> create database backupdb default character set utf8 collate utf8_general_ci;
```

```
# 복원
```

```
# mysql -u root -p (데이터베이스 이름) < (백업 파일)
```

```
$ mysql -u root -p backupdb < 20200412_0203.sql
```

3. 실습 2 : Cold Physical Backup

- <https://dev.mysql.com/doc/mysql-backup-excerpt/5.6/en/innodb-backup.html>

1. 백업 파일 이동

데이터가 저장되는 디렉토리 확인

```
-----  
mysql> show variables like 'datadir';  
/var/lib/mysql/  
-----
```

서버 중지

```
-----  
$ sudo systemctl stop mysql  
-----
```

backup 디렉토리 생성하여 backup 파일을 이동

```
-----  
$ mkdir backup  
$ sudo su  
$ cd /var/lib/mysql  
$ cp -r * /home/ubuntu/backup  
-----
```

backup 파일의 권한을 변경 : 파일 이동시 ubuntu 권한을 가지고 있어야 함

```
-----  
$ sudo chown -R ubuntu: /home/ubuntu/backup
```

Cyberduck 을 이용하여 local pc로 backup 디렉토리 파일 이동

2. 서버 생성

AWS에서 백업한 데이터베이스를 이동할 EC2 서버 생성

3. Mysql 설치

새로 생성한 서버에 Mysql 설치

\$ sudo apt-get update

\$ sudo apt-get upgrade

\$ sudo apt-get install -y mysql-server mysql-client

패스워드 및 설정 파일 설정

\$ sudo mysql

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'rada';

mysql> FLUSH PRIVILEGES;

mysql> exit;

\$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

- bind-address = 0.0.0.0

\$ sudo mysql -u root -prada

mysql> grant all privileges on *.* to root@'%' identified by 'rada';

```
mysql> exit;
```

```
# 서버 재시작하여 설정 적용
```

```
$ sudo service mysql restart
```

```
# mysql workbench로 접속이 잘되는지 확인
```

```
# mysql 서버 중지 : cold backup 을 위한 서비스 중지
```

```
$ sudo service mysql stop
```

4. 백업 파일 이동

Cyberduck 을 이용하여 local pc에 있는 backup 디렉토리 파일을 서버로 이동

```
# 원래 있던 DB 파일 임시 디렉토리로 이동
```

```
$ mkdir tmp
```

```
$ sudo su
```

```
$ cd /var/lib/mysql
```

```
$ cp -r * /home/ubuntu/tmp
```

백업 DB 파일을 이동

\$ cp -r /home/ubuntu/backup/* ./

백업 DB의 모든 파일의 권한을 mysql로 변경

\$ sudo chown -R mysql: /var/lib/mysql

mysql 서비스 시작

\$ sudo service mysql start

접속 후 확인 : 접속할때는 백업 받은 MYSQL의 패스워드로 접속해야 함