Percona xtrabackup

\*\* green is for the command

\*\* blue is for the example

Manual full/incremental backup

Percona xtrabackup is when you transfer the tablespace from source to target.

Condition:

* Make sure percona is installed for both backup and restoration server.
* Sufficient space in server.

1. Run full backup

* Create a directory for full back backup

mkdir -p /path/to/full/backup

mkdir -p /backups/db\_backup/full

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

* Run percona full backup command

xtrabackup --backup \

--target-dir=/path/to/full/backup \

--user=mysql\_username \

--password='mysql\_password’

sudo xtrabackup --backup \

--target-dir=/backups/db\_backup/full \

--user=backup\_user \

--password='Backup@123'

1. Run incremental backup (if there is an incremental backup after the full backup)

* Create a directory for incremental backup

mkdir -p /path/to/incremental/backup

mkdir -p /backups/db\_backup/inc

* Run percona incremental backup command

xtrabackup –backup \

--target-dir=/path/to/incremental/backup \

--incremental-basedir=/path/to/full/backup \

--user=mysql\_username \

--password=’mysql\_password’

sudo xtrabackup --backup \

--target-dir=/backups/db\_backup/inc \

--incremental-basedir=/backups/db\_backup/full \

--user=backup\_user \

--password='Backup@123'

\*\*\*\*\*\* Optional if to save backup and restore in another server\*\*\*\*

rsync -avz /path/to/backup/ server\_username@ip\_adress:/path/to/backup

path/to/backup/ is the path which the full and incremental is saved.

sudo rsync -avz /backups/db\_backup/ sabsystem@10.28.56.210:/backups/db\_backup/

1. Prepare the backup for restoration (use the server you want to restore in). Must prepare from full backup. Then supplement with incremental backup. Incremental backup must be in order. From the first incremental to second then the subsequent.

┌─────────────────────┐

│ Full Backup │ <-- (Prepare fullbackup first)

│ (All Data) │

└─────────────────────┘

|

| Changes only

▼ (Incrementals)

┌─────────────┐ ┌─────────────┐ ┌─────────────┐

│Incremental 1 │ │Incremental 2 │ | Incremental N |

│(Changed Data │🡪 │(Changed Data │🡪 | (changed data |

│since Full) │ │since Inc.1) │ | since N-1) |

└─────────────┘ └─────────────┘ └─────────────┘

----------------------------------🡪 incremental backup must be prepared from first then second then subsequently

* Command for full

xtrabackup --prepare \

--apply-log-only \

--target-dir=/path/to/full/backup

sudo xtrabackup --prepare \

--apply-log-only \

--target-dir=/backups/db\_backup/full

* Command for incremental 1

xtrabackup --prepare \

--apply-log-only \

--target-dir=/path/to/full/backup \

--incremental-dir=/path/to/incremental/backup

sudo xtrabackup --prepare \

--apply-log-only \

--target-dir=/backups/db\_backup/full \

--incremental-dir=/backups/db\_backup/inc

* Command for incremental N+1

xtrabackup --prepare \

--apply-log-only \

--target-dir=/path/to/full/backup \

--incremental-dir=/path/to/incremental\_N+1/backup

xtrabackup --prepare \

--apply-log-only \

--target-dir=/backup/db\_backup/full \

--incremental-dir=/backup/db\_backup/inc2

1. Finalize the backup preparation for restoration

xtrabackup --prepare --target-dir=/path/to/full/backup

sudo xtrabackup --prepare --target-dir=/backups/db\_backup/full

1. Stop mysql services

sudo systemctl stop mysql

sudo systemctl status mysql



1. Clear out old mysql data and create new directory with same name

rm -rf /var/lib/mysql/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*OPTIONAL (to save old data)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sudo mv /var/lib/mysql /path/to/save

sudo mkdir var/lib/mysql\_old\_data

sudo mv /var/lib/mysql /var/lib/mysql\_old\_data

sudo mkdir /var/lib/mysql

1. Paste the prepared backup into var/lib/mysql

sudo cp -avr /path/to/full/backup/\* /var/lib/mysql/

sudo cp -avr /backups/db\_backup/full/\* /var/lib/mysql/

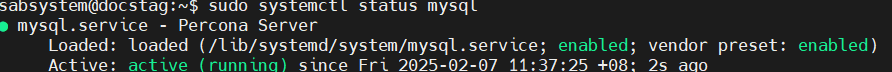
1. Give permission and ownership to the new var/lib/mysql to mysql

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

1. Start mysql

sudo systemctl start mysql

sudo systemctl status mysql



Manual restoration for single table

Conditions:

* Percona installed for both backup and restoration server
* The restoration server must have the same environment such as metadatafiles, tablespace, etc. Senang cite dua2 server kena sama kecuali database.

1. Run a compressed full backup

xtrabackup --backup --compress --compress-threads=8 --target-dir=/path/to/backup

xtrabackup --backup --compress --compress-threads=8 --target-dir=/root/backupdir

1. Copy table to temp for single table extract

mkdir /tmp/anyname

mkdir /tmp/restore

cp -R =/path/to/backup/ /tmp/anyname

cp -R /root/backupdir/\* /tmp/restore

1. Decompress the backup

xtrabackup --decompress --target-dir=/tmp/anyaname --remove-original

xtrabackup --decompress --target-dir=/tmp/restore --remove-original

* Check the table in source server

cd /tmp/anyname/database\_name/

cd /tmp/restore/test/

ls -lhtr ---Check the directory for all the tables inside the database---

1. prepare to import with -export

xtrabackup --prepare --export --target-dir=/tmp/anyaname

xtrabackup --prepare --export --target-dir=/tmp/restore

1. \*\*\*Optional if restoration in another server\*\*\*\*

* Create directory in destination server

mkdir -p /tmp/anyname/database\_name

mkdir -p /tmp/restore/test

* Give ownership of the directory to user you have access to

sudo chown -R user:user /tmp/anyname/database\_name

sudo chown -R sabsystem:sabsystem /tmp/restore/test

* Transfer the directory from source server to destination server

rsync -avz /tmp/anyname/database\_name user@ip\_address:/tmp/anyname/database\_name

\*\*Note: 1st path is based on the path the backup of one database is saved on the source. The 2nd path is the path that you have created above.

rsync -avz /tmp/restore/test sabsystem@10.28.56.210:/tmp/restore/test

1. Create the table being restored in the destination server. (masuk mysql)

* Login mysql

mysql -u user -p

mysql -u SAB -p

* Delete the table being restored. \*\*\* Optional if trying to simulate\*\*\*

Use database\_the\_table\_resides;

Use test;

DROP TABLE table\_to\_restore;

DROP TABLE test1;

* Create table being restore

Create table whatever (

Column1 varchar(10),

Column2 int);

create table test1(

id INT primary key,

name varchar(255),

age int);

* Discard the table tablespace;

alter table whatever discard tablespace;

alter table test1 discard tablespace;

1. Restore the table into destination server

* Stop mysql

sudo systemctl stop mysql

* Transfer the table data to var/lib/mysql

cp /tmp/anyname/database\_name

/var/lib/mysql/database\_name/

cp /tmp/restore/test/test/\* /var/lib/mysql/test/

* Check the data being transferred

ls -lhtr /var/lib/mysql/database\_name/

ls -lhtr /var/lib/mysql/test/

* Give permission to mysql

sudo chown - R mysql:mysql /var/lib/mysql/database\_name/

sudo chown - R mysql:mysql /var/lib/mysql/test

* import table space (masuk balik mysql)

alter table whatever import tablespace;

alter table test1 import tablespace;

* Start mysql again

Sudo systemctl start mysql

Automate percona backup

Conditions:

* Must have python 3.
* Automate percona backup in source server and send the backup file to source.
* Automate the restore in target server with backup file from source server.