Percona xtrabackup

** green is for the command

** blue is for the example

FOR GIT THE PASSWORD SENSITIVE INFO HAS BEEN CHANGED

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=’password’
      ```

2. 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=’password’
```

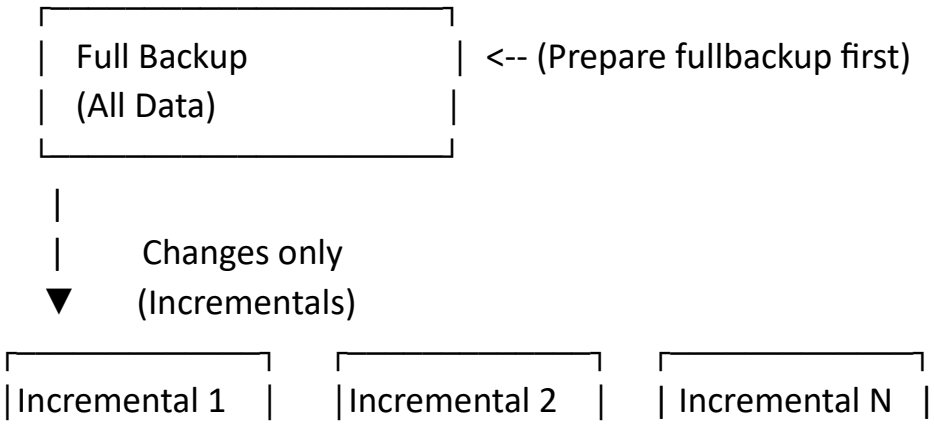
****** Optional if to save backup and restore in another server****

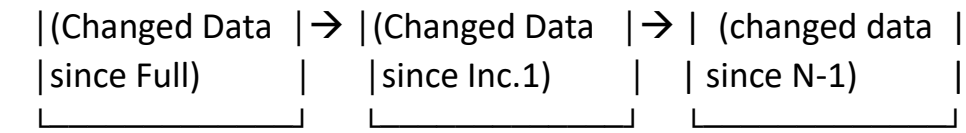
```
rsync -avz /path/to/backup/
server_username@ip_adress:/path/to/backup
```

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

```
sudo rsync -avz /backups/db_backup/
haziq@0.0.0.0:/backups/db_backup/
```

3. 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.





----------→ 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
```

4. Finalize the backup preparation for restoration

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

   sudo xtrabackup --prepare --target-dir=/backups/db_backup/full

5. Stop mysql services

   sudo systemctl stop mysql

   sudo systemctl status mysql

```
● mysql.service - Percona Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: inactive (dead) since Fri 2025-02-07 11:28:52 +08; 17s ago
```

6. 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

7. 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/

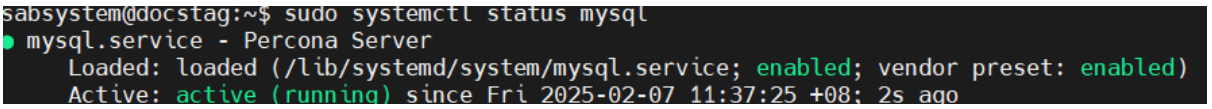
8. Give permission and ownership to the new var/lib/mysql to mysql

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

9. Start mysql

   sudo systemctl start mysql

   sudo systemctl status mysql


sabsystem@docstag:~$ sudo systemctl status mysql
mysql.service - Percona Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2025-02-07 11:37:25 +08; 2s ago

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

2. 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

3. 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---

4. prepare to import with -export
   xtrabackup --prepare --export --target-dir=/tmp/anyaname

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

5. ***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 haziq:haziq /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 haziq@0.0.0.0:/tmp/restore/test

6. Create the table being restored in the destination server. (masuk mysql)
   - Login mysql
     mysql -u user -p

     mysql -u haziq -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;

7. 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.

```
#!/usr/bin/env python3

import os

import sys

import argparse

import subprocess

import logging

from datetime import datetime, timedelta


# ——————————————————————————————————————————————

# CONFIGURATION

# ——————————————————————————————————————————————

# You can also load these from environment variables or a config file.

MYSQL_USER     = 'backup_user'

MYSQL_PASSWORD = 'Backup@123'

BACKUP_BASE    = '/backups/db_backup'

FULL_RETENTION_DAYS = 14


# ——————————————————————————————————————————————

# LOGGER SETUP
```

```
# ——————————————————————————————————————
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s [%(levelname)s] %(message)s',
    handlers=[logging.StreamHandler(sys.stdout)]
)
logger = logging.getLogger()

# ——————————————————————————————————————
# HELPERS
# ——————————————————————————————————————
def run_command(cmd):
    """Run shell command, raise on error."""
    logger.info('Running: %s', ' '.join(cmd))
    result = subprocess.run(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, text=True)
    if result.returncode != 0:
        logger.error('Error: %s', result.stderr.strip())
        raise RuntimeError(f"Command failed: {' '.join(cmd)}")
    logger.info(result.stdout.strip())
    return result.stdout

def rotate_old_backups(path, keep_days):
    """Delete backup directories older than keep_days."""
    cutoff = datetime.now() - timedelta(days=keep_days)
    for name in os.listdir(path):
        fullpath = os.path.join(path, name)
```

```
        if os.path.isdir(fullpath):
            # Expect folder names like YYYYMMDD_HHMMSS
            try:
                ts = datetime.strptime(name, '%Y%m%d_%H%M%S')
            except ValueError:
                continue
            if ts < cutoff:
                logger.info('Removing old backup: %s', fullpath)
                subprocess.run(['rm', '-rf', fullpath])

# ——————————————————————————————————————————————————————
# BACKUP FUNCTIONS
# ——————————————————————————————————————————————————————
def full_backup():
    ts = datetime.now().strftime('%Y%m%d_%H%M%S')
    target = os.path.join(BACKUP_BASE, 'full', ts)
    os.makedirs(target, exist_ok=True)

    cmd = [
        'xtrabackup',
        '--backup',
        f'--target-dir={target}',
        f'--user={MYSQL_USER}',
        f'--password={MYSQL_PASSWORD}',
    ]
    run_command(cmd)
```

```
    # prepare the full backup for restores
    run_command(['xtrabackup', '--prepare', f'--apply-log-only', f'--target-dir={target}'])
    rotate_old_backups(os.path.join(BACKUP_BASE, 'full'), FULL_RETENTION_DAYS)
    logger.info('Full backup completed: %s', target)

def incremental_backup():
    # find latest full backup
    full_dir = os.path.join(BACKUP_BASE, 'full')
    latest_full = sorted(os.listdir(full_dir))[-1]
    base_dir = os.path.join(full_dir, latest_full)

    ts = datetime.now().strftime('%Y%m%d_%H%M%S')
    inc_target = os.path.join(BACKUP_BASE, 'inc', ts)
    os.makedirs(inc_target, exist_ok=True)

    cmd = [
        'xtrabackup',
        '--backup',
        f'--target-dir={inc_target}',
        f'--incremental-basedir={base_dir}',
        f'--user={MYSQL_USER}',
        f'--password={MYSQL_PASSWORD}',
    ]
    run_command(cmd)
```

```
    rotate_old_backups(os.path.join(BACKUP_BASE, 'inc'),
FULL_RETENTION_DAYS)
    logger.info('Incremental backup completed: %s (base: %s)', inc_target,
base_dir)


# ——————————————————————————————————————————————————————————

# CLI

# ——————————————————————————————————————————————————————————

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Percona XtraBackup
automation')
    group = parser.add_mutually_exclusive_group(required=True)
    group.add_argument('--full', action='store_true', help='Run full backup')
    group.add_argument('--incremental', action='store_true', help='Run
incremental backup')
    args = parser.parse_args()


    try:
        if args.full:
            full_backup()
        else:
            incremental_backup()
    except Exception as e:
        logger.exception('Backup failed: %s', e)
        sys.exit(1)
```

THIS IS CRON TO AUTOMATE THE SCRIPT

Full backup every day at 02:00

0 2 * * * /usr/bin/python3 /path/to/backup.py --full >> /var/log/xtrabackup-full.log 2>&1

Incremental backup at the top of every hour

0 * * * * /usr/bin/python3 /path/to/backup.py --incremental >> /var/log/xtrabackup-inc.log 2>&1

If You want to restore use the backup script.