

Bab 8

Melakukan Backup Data dengan mysqldump

Menentukan nama pengguna dan kata sandi

MySQL adalah sistem manajemen basis data relasional open-source yang paling populer. MySQL Server memungkinkan kita untuk membuat banyak akun pengguna dan memberikan hak istimewa yang sesuai sehingga pengguna dapat mengakses dan mengelola basis data.

MySQL adalah sistem manajemen basis data relasional open-source yang paling populer. MySQL Server memungkinkan kita untuk membuat banyak akun pengguna dan memberikan hak istimewa yang sesuai sehingga pengguna dapat mengakses dan mengelola basis data.

Untuk mengakses shell MySQL, ketik perintah berikut dan masukkan kata sandi pengguna root MySQL Anda ketika diminta:

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

Jika Anda belum atau tidak menetapkan kata sandi untuk pengguna root MySQL, Anda dapat menghilangkan flag -p. Jika Anda belum atau tidak menetapkan kata sandi untuk pengguna root MySQL, Anda dapat menghilangkan flag -p.

Akun pengguna di MySQL terdiri dari nama pengguna dan hostname. Untuk membuat akun pengguna MySQL baru, jalankan perintah berikut:

```
CREATE USER 'nama_user'@'localhost' IDENTIFIED BY 'password_user';
```

Ganti nama_user dengan nama pengguna baru yang Anda maksudkan, dan user_password dengan kata sandi pengguna. Pada perintah di atas bagian hostname diatur ke localhost yang berarti bahwa pengguna akan dapat terhubung ke server MySQL hanya dari localhost (Yaitu dari sistem di mana MySQL Server berjalan).

Untuk memberikan akses dari host lain, ubah bagian hostname (localhost) dengan IP mesin jarak jauh. Misalnya, untuk memberikan akses dari mesin dengan IP 10.5.0.4 Anda akan menjalankan:

```
CREATE USER 'user_database'@'10.5.0.4' IDENTIFIED BY 'password_user';
```

Untuk membuat pengguna yang dapat terhubung dari host mana pun, gunakan wildcard '%' sebagai bagian host:

```
CREATE USER 'user_database'@'%' IDENTIFIED BY 'password_user';
```

Opsi ini biasanya digunakan oleh para webmaster yang menginginkan MySQL server ditempat terpisah dengan web server.

Berikan Hak Akses Istimewa ke Akun Pengguna MySQL. Ada beberapa jenis hak istimewa yang dapat diberikan ke akun pengguna. Anda dapat menemukan daftar lengkap hak istimewa yang didukung oleh MySQL di sini.

Hak istimewa yang paling umum digunakan adalah:

- ALL PRIVILEGES - memberikan semua hak istimewa ke akun pengguna.
- CREATE - akun pengguna diizinkan untuk membuat database dan tabel.
- DROP - akun pengguna diizinkan untuk menghapus database dan tabel.
- DELETE - akun pengguna diizinkan untuk menghapus baris dari tabel tertentu.
- INSERT - akun pengguna diizinkan untuk memasukkan baris ke tabel tertentu.
- SELECT - akun pengguna diizinkan untuk membaca database.
- UPDATE - akun pengguna diizinkan untuk memperbarui baris tabel.

Untuk memberikan hak khusus ke akun pengguna, Anda dapat menggunakan sintaks berikut:

```
GRANT      permission1,      permission2      ON
nama_database.nama_tabel      TO
'user_database'@'localhost';
```

Berikut ini beberapa contohnya:

- Memberikan semua hak istimewa ke akun pengguna untuk database tertentu :

```
GRANT ALL PRIVILEGES ON nama_database.* TO
'user_database'@'localhost';
```

- Memberikan semua hak istimewa ke akun pengguna untuk semua database :

```
GRANT ALL PRIVILEGES ON *.* TO
'user_database'@'localhost';
```

- Berikan semua hak istimewa ke akun pengguna untuk tabel tertentu dari database:

```
GRANT ALL PRIVILEGES ON nama_database.nama_tabel TO
'user_database'@'localhost';
```

- Berikan hanya beberapa hak istimewa ke akun pengguna untuk database tertentu:\

```
GRANT SELECT, INSERT, DELETE ON nama_database.* TO
user_database@'localhost';
```

Tampilkan Hak Akses User MySQL. Untuk menemukan hak istimewa yang diberikan kepada akun pengguna MySQL tertentu, gunakan pernyataan SHOW GRANTS:

```
SHOW GRANTS FOR 'user_database'@'localhost';
```

```
+-----+
+-----+
| Grants for database_user@localhost |
+-----+
+-----+
| GRANT USAGE ON *.* TO 'database_user'@'localhost' |
+-----+
```

```
| GRANT ALL PRIVILEGES ON `database_name`.* TO  
'database_user'@'localhost' |
```

```
+-----+  
-----+
```

2 rows in set (0.00 sec)

Revoke (Cabut) Hak Istimewa dari Akun Pengguna MySQL. Sintaks untuk mencabut satu atau lebih hak istimewa dari akun pengguna hampir sama dengan ketika memberikan hak istimewa.

Misalnya untuk mencabut semua hak istimewa dari akun pengguna untuk database tertentu, gunakan perintah berikut:

```
REVOKE ALL PRIVILEGES ON nama_database.* TO  
'user_database'@'localhost';
```

Hapus Akun Pengguna MySQL. Untuk menghapus akun pengguna MySQL, gunakan pernyataan DROP USER:

```
DROP USER 'user_database'@'localhost'
```

Perintah di atas akan menghapus akun pengguna dan hak istimewanya.

Membuat Cadangan Database atau Tabel

Mysqldump adalah salah satu fasilitas bawaan server MYSQL, yang berfungsi untuk membackup database mysql. mysqldump adalah perintah command line, tapi apabila menghendaki dijalankan rutin bisa di pasang di cron system.

Contoh penggunaan :

```
mysqldump -user [user name]-password=[password]  
[database name] > [dump file]
```

Keterangan perintah di atas :

-user [user name] : user name adalah user account dalam MYSQL server

`-password=[password]` : password dari user name
`[database name]` : nama database yang akan di backup ,
apabila kita ingin membackup semua database yang ada
gunakan options `-all-databases`
`[dump file]` : adalah nama file hasil backup database,
bisa gunakan path , misal `/var/backup/mysql.dump`

Selain bisa digunakan membackup database ke file,
mysqldump bisa juga membackup database ke database
mysql di server lain. dengan perintah : `mysqldump -`
`user admin -password=test -opt mydata | mysql -`
`host=192.168.13.13 -C newmydata`

Keterangan :

`-opt` : options ini berfungsi mengoptimalkan fungsing
pembacaan database

`-host` : server tujuan tempat data base tujuan (di
server tujuan sudah dibuat newmydata database terlebih
dahulu)

`-C` : options ini berfungsi agar data di kompresi
terlebih dahulu sebelum di kirim ke database tujuan

Category: database Tags: backup, database, mysql,
mysqldump

Opsi-opsi Pencadangan:

- **Cadangkan semua basis data**

Opsi ini `-all-databases` memungkinkan untuk mencadangkan semua basis data.

```
mysqldump -u root -p --all-databases > backup.sql
```

- **Cadangkan hanya database tertentu**

Opsi ini `-databases` memungkinkan untuk mencadangkan hanya database tertentu dengan memberikan nama-namanya ke command line.

```
mysqldump -u root -p --databases market company > backup.sql
```

- **Cadangkan satu basis data**

Opsi ini `-databases` memungkinkan untuk membuat cadangan satu database dengan memberikan nama database ke command line.

```
mysqldump -u root -p --databases market > backup.sql
```

Opsi `-databases` ini dapat dihilangkan jika ingin membuat cadangan satu database.

```
mysqldump -u root -p market > backup.sql
```

Perbedaan antara kedua perintah ini adalah, tanpa `-databases` opsi, cadangan tidak memiliki `CREATE DATABASE` dan pernyataan `USE`.

- **Cadangkan hanya tabel tertentu dari database**

Berikan nama database dan nama tabel ke command line tanpa `-databases` opsi untuk hanya mencadangkan tabel tertentu dari database.

```
mysqldump -u root -p market products  
manufacturers > backup.sql
```

- **Cadangkan struktur basis data tanpa data**

Opsi ini `-no-datamemungkinkan` untuk membuat cadangan struktur database tanpa data. Jadi file cadangan hanya berisi pernyataan untuk membuat tabel.

```
mysqldump -u root -p --no-data market >  
backup.sql
```

- **Backup data database tanpa struktur**

Opsi ini `-no-create-info` memungkinkan untuk membuat cadangan data database tanpa struktur. Jadi file cadangan hanya berisi data tabel.

```
mysqldump -u root -p --no-create-info market  
> backup.sql
```

- **Cadangkan database tanpa mengunci**

- Secara default, semua tabel akan dikunci hingga pencadangan selesai. Opsi ini `-single-transaction` memungkinkan untuk membuat cadangan basis data tanpa mengunci tabel.

```
mysqldump -u root -p --single-transaction  
market > backup.sql
```

Ketika opsi ini digunakan, cara isolasi transaksi diatur ke `REPEATABLE READ` dan `START TRANSACTION` pernyataan dikirim ke server sebelum memulai pencadangan basis data.

Untuk memastikan pencadangan berhasil, koneksi lain tidak dapat menggunakan pernyataan berikut: `ALTER TABLE`, `CREATE TABLE`, `DROP TABLE`, `RENAME TABLE`, `TRUNCATE TABLE`.

Opsi ini `-single-transaction` hanya berfungsi dengan tabel transaksional, seperti InnoDB. Opsi ini tidak akan bekerja dengan tabel MyISAM dan MEMORY. Tabel ini tidak bersifat transaksional, sehingga status dapat diubah selama pencadangan.

- **Cadangan database besar**

Berguna untuk menggabungkan `-single-transaction` opsi dengan `-quick` opsi ketika perlu membuat cadangan database besar.

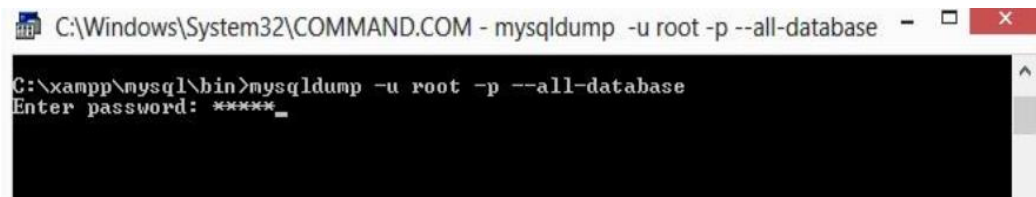
```
mysqldump -u root -p --single-transaction --  
quick market > backup.sql
```

Saat opsi `-quick` digunakan, data diambil satu per satu baris daripada mengambil seluruh rangkaian baris dan menyimpannya di memori.

Memulihkan Cadangan Database atau Tabel

Tool `MySQLdump` memiliki kelebihan dalam proses backup database. Tools ini akan melakukan backup database beserta struktur table di dalamnya. Selain itu, file backup yang dihasilkan dapat digunakan bagi sistem database yang lain. Langkah-langkah penggunaan `MySQLdump` dalam proses backup dan recovery database adalah sebagai berikut :

- Jalankan shell atau command-prompt dan ketikkan perintah berikut untuk memulai dump database:
`MySQLDUMP -u root -p -all-database`



Perintah di atas akan melakukan backup pada semua database yang ada pada MySQL.

- Proses backup atau dumping akan berlangsung beberapa saat hingga muncul pesan "Dump completed".

```
> ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table 'user_pwd'
--

LOCK TABLES 'user_pwd' WRITE;
/*!40000 ALTER TABLE 'user_pwd' DISABLE KEYS */;
INSERT INTO 'user_pwd' VALUES ('xampp','wampp');
/*!40000 ALTER TABLE 'user_pwd' ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2013-11-26 21:36:03
```

- Data yang telah di-backup dapat di restore kembali ke dalam database dengan perintah:
MySQLdump -u root -p (nama_database) < c:\file_backup.sql

Mentransfer Data Dari Satu Server

Jika Anda menggunakan Database Migration Service API untuk membuat tugas migrasi, pastikan Anda menggunakan mysqldump utilitas untuk memperbarui dump saat menjalankan panggilan REST API. Untuk informasi selengkapnya, lihat Mengelola tugas migrasi menggunakan API.

Anda dapat menjalankan utilitas mysqldump langsung terhadap database MySQL Anda, menggunakan opsi apa pun yang Anda butuhkan. Namun, jika Anda mengekspor untuk mengimpor data ke database Cloud SQL, gunakan utilitas mysqldump dengan tanda berikut:

- `--databases` Tentukan daftar eksplisit database yang akan diekspor. Daftar ini tidak boleh berisi database sistem (`sys`, `mysql`, `performance_schema`, dan `information_schema`).
- `--hex-blob` Jika database Anda berisi bidang biner apa pun, maka Anda harus menggunakan tanda ini untuk memastikan bahwa bidang biner Anda diimpor dengan benar.
- `--single-transaction` Memulai transaksi sebelum dijalankan. Daripada mengunci seluruh database, ini memungkinkan mysqldump membaca database dalam keadaan saat ini, membuat dump data yang konsisten..
- `--routines` Untuk memasukkan prosedur dan fungsi yang tersimpan.

Disarankan juga untuk menggunakan flag berikut:

- `--no-autocommit`
- `--default-character-set=utf8mb4`
- `--master-data`

Saat mysqldump sedang berjalan, jangan lakukan operasi DDL apa pun pada database. Melakukannya dapat menyebabkan inkonsistensi dalam file ekspor.

Dari mesin dengan konektivitas jaringan ke server MySQL Anda, jalankan perintah berikut:

```
mysqldump \
  -h [SOURCE_ADDR] -P [SOURCE_PORT] -u
[USERNAME] -p \
  --databases [DBS] \
  --hex-blob \
  --no-autocommit \
  --default-character-set=utf8mb4 \
  --master-data=1 \
  --single-transaction \
  --routines \
  | gzip \
```

```

| gsutil cp -
gs://[BUCKET_NAME]/[DUMP_FILENAME].gz

```

Jika server database sumber Anda tidak mendukung GTID, gunakan `--master-data=1` properti; jika tidak, jangan gunakan properti ini. Juga, jika server database sumber Anda mendukung GTID, gunakan `--set-gtid-purged=on` properti; jika tidak, jangan gunakan properti ini.

Jika sumber migrasi adalah Relational Database Service (RDS) untuk MySQL:

- Properti `master-data` tidak didukung.
- Jika server database sumber Anda mendukung GTID, gunakan `--set-gtid-purged=on` properti; jika tidak, jangan gunakan properti ini.
- Jika Anda menggunakan dump manual untuk memigrasikan data, lakukan migrasi dengan mengaktifkan GTID.

Perintah ini mungkin terlihat seperti contoh berikut:

```

mysqldump \
  -h [SOURCE_ADDR] -P [SOURCE_PORT] -u
[USERNAME] -p \
  --databases [DBS] \
  --hex-blob \
  --no-autocommit \
  --default-character-set=utf8mb4 \
  --set-gtid-purged=on \
  --single-transaction \
  --routines \
  | gzip \
  | gsutil cp -
gs://[BUCKET_NAME]/[DUMP_FILENAME].gz

```

Selain itu, Anda harus mengonfigurasi instans RDS untuk mempertahankan binlog lebih lama. Perintah ini mungkin terlihat seperti contoh berikut:

```

# Sets the retention period to one week.
call mysql.rds_set_configuration('binlog retention
hours', 168);

```

Mengganti [PROPERTIES_IN_BRACKETS] dengan nilai-nilai berikut:

Properti	Nilai
[SOURCE_ADDR]	Alamat IPv4 atau nama host untuk server database sumber.
[SOURCE_PORT]	Port untuk server database sumber.
[USERNAME]	Akun pengguna MySQL.
[DBS]	Daftar database yang dipisahkan oleh spasi di server database sumber untuk disertakan dalam dump. Gunakan SHOW DATABASES perintah MySQL untuk membuat daftar database Anda.
[BUCKET_NAME]	Bucket di Cloud Storage yang dibuat oleh pengguna dan digunakan untuk menyimpan file dump (misalnya, replica-bucket).
[DUMP_FILENAME]	Nama file dump, diakhiri dengan ekstensi file (misalnya, ..gzsource-database.sql.gz

Perintah gzip menyebabkan file dump dikompresi. Jika database Anda berisi data yang tidak terkompresi dengan baik, seperti data biner yang tidak dapat dimampatkan atau gambar JPG, maka hapus | gzip dari perintah.

MySQL Ke Server MySQL Lainnya

Anda juga dapat mengimpor data dari basis data MariaDB atau MySQL ke instans DB MySQL atau MariaDB. Anda melakukannya dengan menyalin database dengan mysqldump dan pipa langsung ke MariaDB atau MySQL DB misalnya. Parameter mysqldump utilitas baris perintah umumnya digunakan untuk membuat pencadangan dan mentransfer data dari satu MariaDB atau MySQL server ke server lainnya. Ini disertakan dengan perangkat lunak klien MySQL dan MariaDB.

Sebuah khasmysqldumpperintah untuk memindahkan data dari basis data eksternal ke instans DB MySQL RDS terlihat mirip dengan yang berikut ini.

```
mysqldump -u local_user \  
--databases database_name \  
--single-transaction \  
--compress \  
--order-by-primary \  
-plocal_password | mysql -u RDS_user \  
--port=port_number \  
--host=host_name \  
-pRDS_password
```

Pastikan Anda menyadari rekomendasi dan pertimbangan berikut ini:

- Jangan sertakan skema berikut dalam file dump: sys, performance_schema, dan information_schema. Utilitas mysqldump tidak menyertakan skema tersebut secara default.
- Jika Anda harus memigrasikan pengguna dan keistimewaan, pertimbangkan untuk menggunakan sebuah alat yang menghasilkan bahasa kontrol data (DCL) untuk menciptakan ulang, seperti utilitas pt-show-grants.
- Untuk melakukan impor, pastikan pengguna melakukannya memiliki akses ke instans DB.

Parameter yang digunakan adalah sebagai berikut:

- -u local_user - Gunakan untuk menentukan sebuah nama pengguna. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan nama akun pengguna pada basis data MariaDB atau MySQL lokal yang diidentifikasi oleh --databases parameter.
- --databases database_name - Gunakan untuk menentukan nama basis data pada instans MariaDB atau MySQL lokal yang ingin Anda impor ke RDS.

- `--single-transaction` - Gunakan untuk memastikan bahwa semua data yang dimuat dari basis data lokal konsisten dengan satu titik waktu. Jika ada proses lain mengubah data sementaramysqldumpmembacanya, menggunakan parameter ini dapat membantu menjaga integritas data.
- `--compress` - Gunakan untuk mengurangi konsumsi bandwidth jaringan dengan mengompres data dari basis data lokal sebelum mengirimkannya ke RDS.
- `--order-by-primary` - Gunakan untuk mengurangi waktu pemuatan dengan mengurutkan masing-masing tabel data berdasarkan kunci primernya.
- `-plocal_password` - Gunakan untuk menentukan sebuah kata sandi. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u`.
- `-u RDS_user` - Gunakan untuk menentukan sebuah nama pengguna. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan nama akun pengguna pada basis data default untuk instans DB MariaDB yang diidentifikasi oleh instans MySQL MariaDB yang diidentifikasi oleh `-hostparameter`.
- `--port port_number` - Gunakan untuk menentukan port untuk instans DB MariaDB Anda. Secara default, port ini adalah 3306, kecuali jika Anda mengubah nilai saat menciptakan instans.
- `--host host_name` - Gunakan untuk menentukan nama Domain Name System (DNS) dari titik akhir instans DB MySQL RDS, misalnya, `"myinstance.123456789012.us-east-1.rds.amazonaws.com"`. Anda dapat menemukan nilai titik akhir dalam rincian instans di MySQL RDS Management Console.

- `-pRDS_password` - Gunakan untuk menentukan sebuah kata sandi. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u` kedua.
- Pastikan untuk menciptakan prosedur, pemicu, fungsi, atau peristiwa apa pun yang tersimpan secara manual dalam basis data MySQL RDS Anda. Jika Anda memiliki salah satu dari objek ini di basis data yang Anda salin, jangan sertakan saat Anda menjalankan `mysqldump`. Untuk melakukannya, sertakan parameter berikut dengan `mysqldump` perintah: `--routines=0 --triggers=0 --events=0`.

Contoh berikut menyalin basis data sampel `world` pada host lokal ke instans DB MySQL.

Untuk Linux, macOS, atau Unix:

```
sudo mysqldump -u localuser \
  --databases world \
  --single-transaction \
  --compress \
  --order-by-primary \
  --routines=0 \
  --triggers=0 \
  --events=0 \
  -plocalpassword | mysql -u rdsuser \
  --port=3306 \
  --host=myinstance.123456789012.us-east-
1.rds.amazonaws.com \
  -prdspassword
```

Untuk Windows, jalankan perintah berikut dalam sebuah prompt perintah yang telah dibuka dengan mengeklik kanan Command Prompt pada menu program Windows dan memilih Jalankan sebagai administrator:

```
mysqldump -u localuser ^
  --databases world ^
  --single-transaction ^
  --compress ^
  --order-by-primary ^
  --routines=0 ^
```

```
--triggers=0 ^
--events=0 ^
-plocalpassword | mysql -u rdsuser ^
--port=3306 ^
--host=myinstance.123456789012.us-east-
1.rds.amazonaws.com ^
-prdspassword
```

Melakukan Backup dari Server Jarak Jauh dengan Kompresi

Cara Backup Jarak Jauh Database MYSQL dari SSH

- Backup dan restore database MySQL dari ssh - SSH (Secure Shell) merupakan protokol transfer yang dapat digunakan untuk melakukan kontrol server jarak jauh. Dengan adanya SSH, maka transfer data lebih terenkripsi sehingga akan lebih aman dari serangan hacker atau lainnya.

Agar dapat menggunakan SSH, maka Anda harus mengerti command atau syntax-syntax SSH. Karena Anda hanya akan menggunakan terminal console hitam putih tanpa adanya GUI yang tinggal klik-klik saja.

Pada kesempatan kalai ini saya akan membahas mengenai cara backup dan restore database menggunakan SSH.

Mengapa perlu menggunakan SSH ?

Bila Anda memiliki database dengan ukuran yang kecil, maka bisa menggunakan phpMyadmin atau cpanel saja untuk melakukan backup atau restore sudah cukup. Namun hal ini akan sangat akan merepotkan bahwa gagal bila database memiliki ukuran besar hingga bergiga-giga. Oleh karena itu Anda membutuhkan SSH untuk melakukan backup ataupun restore. Perlu diketahui juga, bahwa import database melalui phpMyadmin memiliki limit tertentu sesuai dengan layanan hosting yang Anda gunakan.

Sebelum Anda memulai mengikuti panduan ini, pastikan Anda sudah mengetahui detail akses ke SSH hosting

serta caranya. Silakan menghubungi layanan hosting yang Anda gunakan saat ini apakah sudah support dengan fitur SSH. Karena tidak semua paket hosting memiliki akses ke SSH.

➤ Backup Database menggunakan MYSQLDUMP di SSH

Persiapkan semua yang dibutuhkan untuk melakukan backup database dari ssh.

User-DB :

Nama-DB :

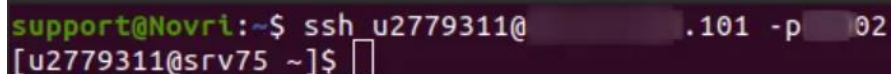
Password-DB :

Pastikan Anda sudah mengetahui detail database tersebut, karena saat backup menggunakan mysqldump akan dibutuhkan.

Informasi lainnya, Pada panduan ini saya mengakses SSH melalui terminal linux, karena laptop saya menggunakan sistem operasi linux. Anda juga dapat melakukan akses dari cpanel >> menu terminal. Selain itu bisa menggunakan aplikasi putty seperti panduan diatas.

- Akses dan login ke SSH.

Gambar dibawah membuktikan bahwa Anda sudah berhasil login ke SSH.

A terminal window with a dark background. The prompt is 'support@Novri:~\$'. The command entered is 'ssh u2779311@... .101 -p ...02'. The output is '[u2779311@srv75 ~]\$' followed by a cursor.

```
support@Novri:~$ ssh u2779311@... .101 -p ...02
[u2779311@srv75 ~]$
```

- Jalankan Perintah MySQLDump

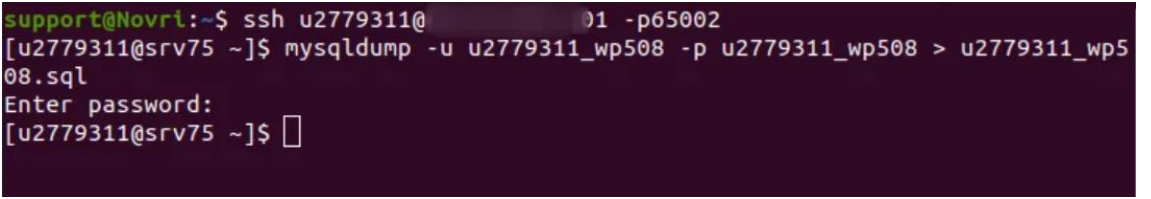
Setelah berhasil login ke SSH. Jalankan command atau perintah dibawah untuk memulai melakukan backup database menggunakan MySQLDump.

```
mysqldump -u user-database -p nama-database > hasil-backup.sql
```

Setelah menjalankan perintah diatas, akan terdapat perintah untuk memasukan password. Silakan Anda masukan "Password Database" dan tekan "Enter". Anda tidak akan mendapatkan notifikasi bahwa backup database telah selesai. Tanda backup selesai akan kembali ke direktori saat pertama kali login ke ssh

atau berhenti dari proses. Ketika ada error akan ada notifikasi dilayar. Waktu yang dibutuhkan untuk proses backup menggunakan MySQLDump biasanya tergantung pada ukuran database. Semakin besar ukurannya maka akan semakin lama pula.

Gambar dibawah membuktikan bahwa proses backup berhasil dan tidak ada error.

A terminal window screenshot showing a successful MySQL backup process. The user 'support@Novri' connects via SSH to 'u2779311@'. They then run the command 'mysqldump -u u2779311_wp508 -p u2779311_wp508 > u2779311_wp508.sql'. The prompt 'Enter password:' is shown, followed by the user's input and the confirmation '[u2779311@srv75 ~]\$'.

- Cek Hasil Backup

Anda dapat melakukan pengecekan hasil backup melalui File Manager ataupun SSH. Anda dapat melakukan pengecekan dari SSH dengan menjalankan perintah dibawah :

Ll

Ketika Anda menjalankan perintah diatas, maka semua file yang ada pada direktori tempat Anda berada akan muncul.

File backup database akan berekstensi .sql seperti screenshot dibawah. Anda dapat mendownload file database.sql melalui file manager cpanel hosti ataupun melalui SSH. Agar lebih mudah bisa Anda langsung download melalui file manager saja.

```

total 2836
lrwxrwxrwx 1 u2779311 u2779311 34 Des 1 2020 access-logs -> /etc/apache2/
logs/domlogs/u2779311
drwxr-x--- 6 u2779311 mail 4096 Okt 13 19:52 etc
drwx----- 3 u2779311 u2779311 4096 Okt 17 07:09 logs
drwx----- 2 u2779311 u2779311 4096 Mar 26 2021 lscmData
drwxr-x--x 13 u2779311 u2779311 4096 Sep 24 00:11 mail
drwxr-x--- 3 u2779311 u2779311 4096 Des 1 2020 public_ftp
drwxr-xr-x 26 u2779311 nobody 4096 Okt 15 15:12 public_html
drwx--x--x 3 u2779311 u2779311 4096 Okt 14 22:15 softaculous_backups
drwxr-xr-x 5 u2779311 u2779311 4096 Okt 13 20:22 ssl
drwxr-xr-x 7 u2779311 u2779311 4096 Okt 13 19:57 tmp
-rw-rw-r-- 1 u2779311 u2779311 2860743 Okt 17 08:14 u2779311 wp508.sql
drwx----- 2 u2779311 u2779311 4096 Mar 25 2021 wordpress-backups
lrwxrwxrwx 1 u2779311 u2779311 11 Des 1 2020 www -> public_html
[u2779311@srv75 ~]$

```

Meskipun ukuran database besar, proses backup menggunakan SSH akan lebih cepat dibandingkan melalui phpMyadmin.

Restore Database menggunakan SSH

Sama seperti proses backup, pastikan Anda sudah menyipkan user database, nama database dan password database.

Akses dan Login SSH

Hal pertama Anda perlu login terlebih dahulu ke SSH dan masuk ke direktori tempat database.sql berada. Pada panduan ini, file database.sql berada di folder public_html. Sehingga saya akan masuk atau pindah ke direktori public_html terlebih dahulu.

```
cd public_html
```

Namun untuk Anda bisa menyesuaikan lagi letak file backup database .sqlnya.

Jalankan Perintah Restore Database

Silakan Anda jalankan perintah berikut ini pada terminal atau command prompt agar proses restore dimulai.

```
mysql -u users-database -p nama-database < nama-filedb.sql
```

Setelah menjalankan perintah diatas, akan terdapat perintah untuk memasukan password. Silakan Anda masukan "Password Database" dan tekan "Enter".

Anda tidak akan mendapatkan notifikasi bahwa restore database telah selesai. Ketika ada error akan ada notifikasi dilayar. Waktu yang dibutuhkan untuk proses restore menggunakan MySQLDump biasanya tergantung pada ukuran database. Semakin besar ukurannya maka akan semakin lama pula.

```
[u2779311@srv75 ~]$ cd public_html/
[u2779311@srv75 public_html]$ mysql -u u2779311_wp508 -p u2779311_wp508 < u2779311_wp508.sql
Enter password:
[u2779311@srv75 public_html]$
```

Anda juga dapat melakukan import langsung database dengan file database.sql berada pada komputer lokal. Namun pada kesempatan kali ini saya hanya akan membahas restore yang mana database.sql sudah Anda upload ke File manager cpanel.

Cek Hasil Restore

Untuk pengecekan database dapat Anda lakukan melalui phpMyadmin hosting. Meskipun pada proses SSH sebelumnya sudah dapat terlihat bahwa proses restore berhasil, karena tidak terdapat pesan error yang muncul.

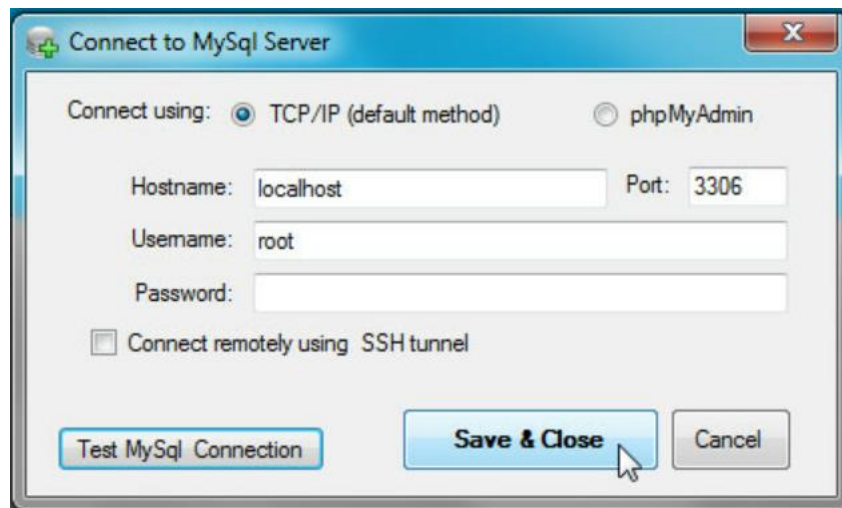
Cara Backup dan Kompres Zip Server Jarak Jauh Menggunakan MySQL Backup FTP

Anda mungkin tahu bahwa alat cadangan DBMS (Database Management System) menawarkan serangkaian opsi terbatas untuk membuat cadangan basis data. Itulah alasan mengapa manajer basis data mengandalkan pihak ketiga, solusi cadangan basis data yang kuat yang dapat mengotomatiskan proses pencadangan semua sistem dan non-sistem database di lokasi yang nyaman. Ambil

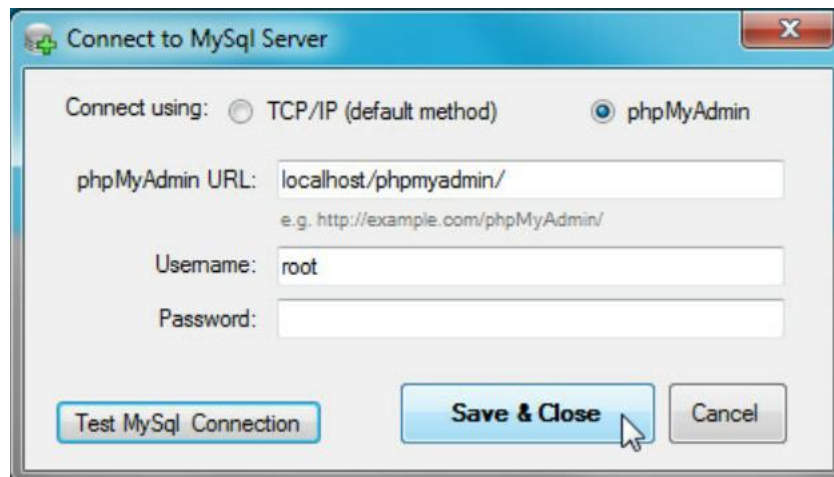
phpmyadmin sebagai contoh, Anda dapat memilih alat Ekspornya untuk membuat cadangan dari basis data Anda, tetapi bagaimana jika Anda ingin membuat cadangan basis data secara teratur, atau setelah setiap pembaruan utama? Karena menjadi membosankan untuk secara manual memilih format output diikuti oleh mode kompresi dari tab Ekspor untuk membuat cadangan, Anda mungkin memerlukan automater cadangan basis data seperti MySQLBackupFTP (a.k.a MySQLBF) untuk secara otomatis melakukan operasi pencadangan database.

Seperti petunjuk nama, MySQLBackupFTP ditulis untuk mencadangkan basis data MySQL dari SQL Server dan alat administrasi MySQL lainnya seperti phpmyadmin, Administrator MySQL, dll. Apa yang membuatnya unik dari aplikasi cadangan database lainnya adalah kemampuannya untuk mengambil dan mengirim basis data langsung ke server FTP, buat cadangan di lokasi lokal atau jaringan, dan untuk menjadwalkan pekerjaan pencadangan untuk secara otomatis membuat cadangan basis data pada waktu yang ditentukan interval.

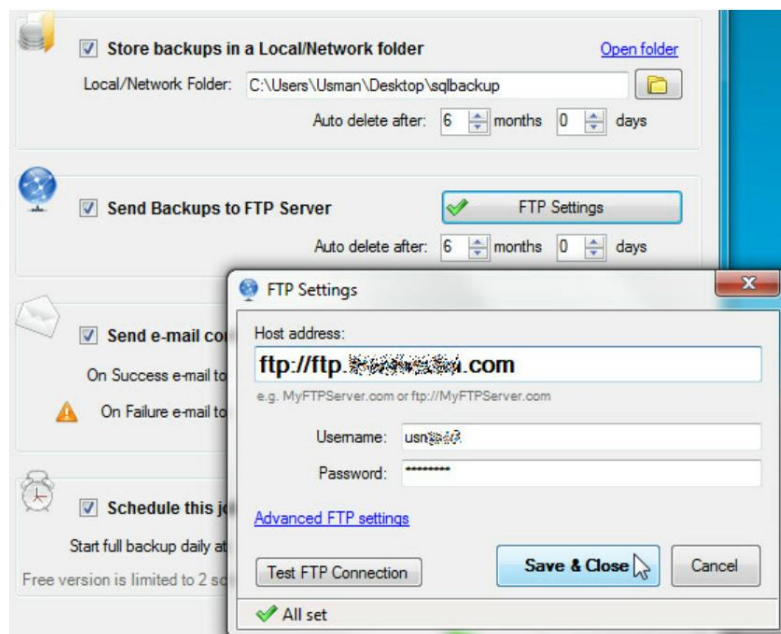
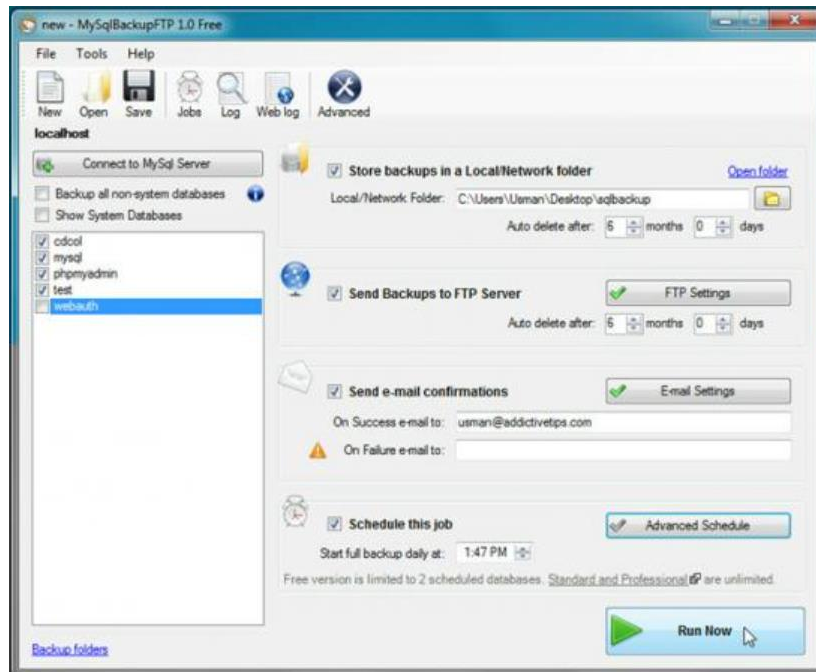
MySQLBackupBF menawarkan dua mode untuk terhubung dengan database MySQL Server; TCP / IP dan phpMyAdmin. Jika Anda bermaksud untuk mencadangkan basis data Server MySQL dari server web lokal (localhost), kami sarankan untuk menggunakan mode TCP / IP untuk terhubung dengan basis data. Sebelum Anda mengklik Simpan dan Tutup untuk membuka antarmuka utama, tekan Test MySql Connection untuk memverifikasi tautan, nama pengguna, dan kata sandi. Untuk membuat koneksi yang aman dengan SQL Server Anda, aktifkan Terhubung dari jarak jauh menggunakan terowongan SSH opsi, masukkan Nama Host, Nama Pengguna, dan Kata Sandi. Pastikan Anda menggunakan port 22 untuk terhubung dengan Server MySQL melalui SSH.



Mode phpMyAdmin mengharuskan Anda memasukkan URL phpmyadmin, Nama Pengguna, dan Kata Sandi. Dalam mode ini, Anda dapat langsung memasukkan jalur sumber alat admin phpmyadmin, seperti `www.somewebsite.com/phpmyadmin` atau gunakan localhost dengan port 8080 standar untuk terhubung dengan phpmyadmin pada server web lokal.

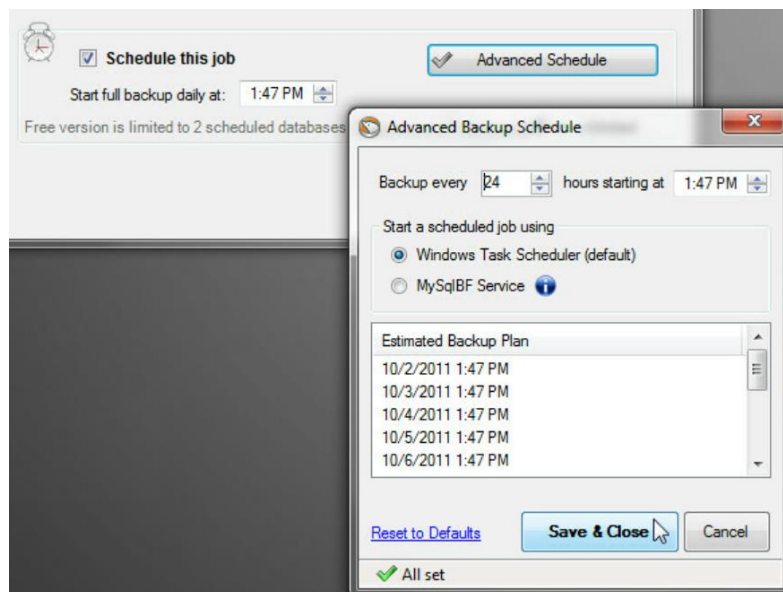


Setelah koneksi dengan MySQL Server dibuat, Anda akan melihat antarmuka utama. Daftar basis data ditampilkan di sisi kiri sementara jendela utama menampung semua opsi dan mode cadangan yang disebutkan di atas. Anda hanya perlu memilih database yang ingin Anda backup dan mengatur opsi backup. MySQLBF memungkinkan Anda untuk menggunakan beberapa mode cadangan juga; Anda dapat memilih lokasi lokal / jaringan dan mengkonfigurasi FTP untuk membuat cadangan di lokasi lokal / jaringan dan server FTP.



Secara default, ini hanya mencadangkan database non-sistem yang penting, tetapi Anda dapat mengaktifkan Show System Database untuk memasukkan semua basis data sistem yang tersedia, seperti skema, DB konfigurasi sistem, dalam cadangan operasi. Setelah Anda memilih database untuk dicadangkan, aktifkan mode cadangan dan opsi dari jendela utama untuk memasukkan jalur sumber dan mengonfigurasi pengaturan yang diperlukan, masing-masing. Simpan cadangan di lokasi lokal / jaringan dan mode server FTP memungkinkan Anda untuk menghapus cadangan secara otomatis setelah beberapa bulan.

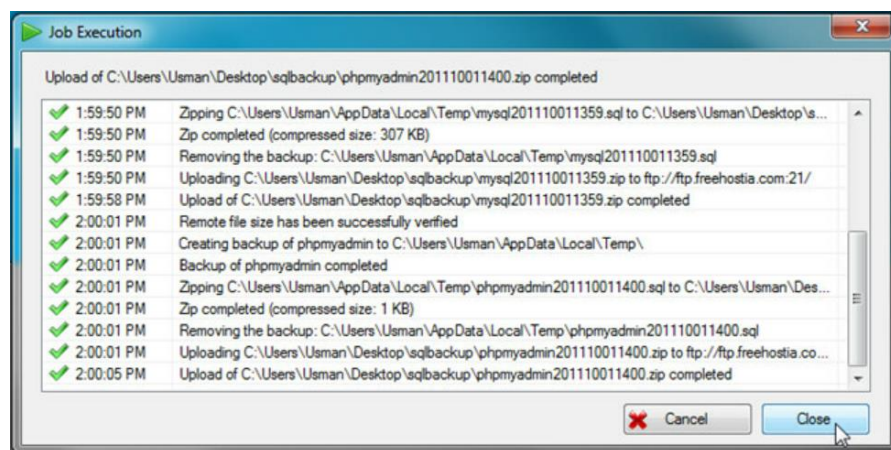
MYSQIBF secara asli mendukung Gmail untuk mengirim log keberhasilan dan kegagalan. Namun, Anda dapat memilih opsi khusus dari Pengaturan Email untuk mengonfigurasi beberapa akun email lainnya. Bermanfaat ketika Anda ingin memberi tahu manajer basis data tentang kegagalan atau keberhasilan tugas cadangan basis data.



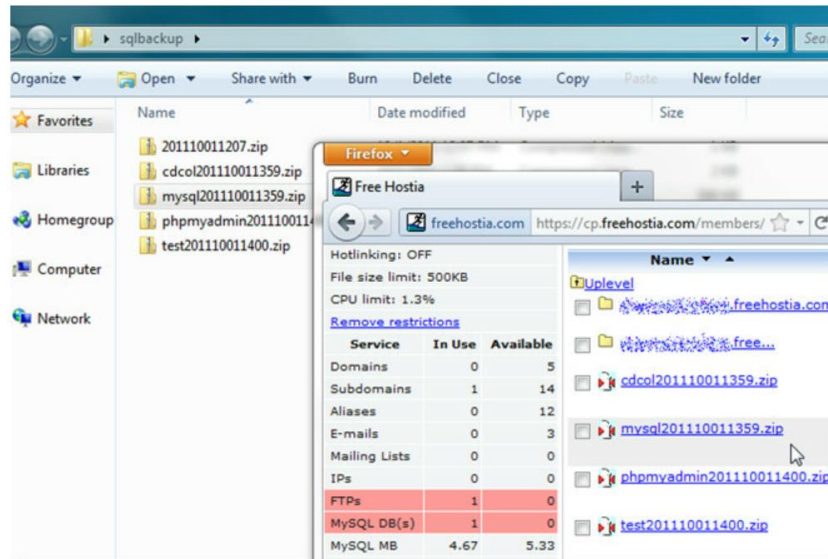
Langkah terakhir adalah menjadwalkan operasi pencadangan. Anda dapat menggunakan mode jadwal

sederhana yang hanya membutuhkan waktu di mana Anda ingin melakukan operasi pencadangan. Sedangkan, mode Jadwal Lanjutan memungkinkan Anda untuk memasukkan jumlah jam setelah itu secara otomatis menjalankan pekerjaan cadangan.

Setelah mode dan opsi cadangan dikonfigurasi, klik Jalankan untuk melakukan operasi cadangan pertama Anda. Ini akan menampilkan log cadangan secara real-time untuk memungkinkan Anda mengidentifikasi dan memperbaiki masalah cadangan database.



Ketika operasi pencadangan selesai, verifikasi database yang dicadangkan di lokasi sumber. Ini akan membuat cadangan basis data dalam format ZIP.



Dianjurkan untuk menyimpan pekerjaan cadangan sebelum Anda menutup aplikasi. MySQLBF menyimpan pekerjaan cadangan dalam format MJOBX miliknya. Harus dicatat bahwa versi gratis dapat membuat jadwal cadangan hanya untuk dua database. Untuk menghapus batasan ini, Anda harus membeli versi Standar (biaya \$ 49) atau versi Profesional (biaya \$ 69). Anda dapat memeriksa perbandingan terperinci antara versi gratis dan versi berbayar di Internet fitur dan harga bagian dari halaman produk. Ia bekerja pada edisi Windows untuk klien dan server.

Mengembalikan File Backup yang di-Gzip Tanpa Membuka Kompresi

Dan adalah utilitas sistem mysqldumpoperasi gzipLinux. Mysqldump Digunakan untuk melakukan backup database. Kami menggunakan yang gzip berikut jika kami dapat menyimpan cadangan sebagai file terkompresi di direktori saat ini.

Contoh Kode Satu:

```
-- Syntax: mysqldump -u username -ppassword db_name |  
gzip > dumpfilename.sql.gz
```

```
mysqldump -u root -p test | gzip > dump_test_db.sql.gz
```

Sekarang, gunakan `ls` perintah untuk mengonfirmasi bahwa file terkompresi telah dibuat. Mari kita lihat contoh lain di mana data melewati file yang berbeda dan disimpan di file terakhir.

Contoh Kode Dua:

```
-- Syntax: mysqldump -u username -ppassword db_name |  
gzip > file1.gz > file2.gz
```

```
mysqldump -u root -p test | gzip > file1.gz > file2.gz  
> file3.gz
```

Keluaran:

```
$> ls -l
```

```
-rw-r--r--  1 root  root      0 14:35 21 May file1.gz
```

```
-rw-r--r--  1 root  root      0 14:35 21 May file2.gz
```

```
-rw-r--r--  1 root  root    453 14:35 21 May file3.gz
```

Dalam contoh ini, `mysqldump` dieksekusi, dan output yang dihasilkan dialihkan menggunakan simbol pipa (`|`). Kemudian, pipa mengirimkan output standar ke dalam `gzip` perintah sebagai input standar. Ini `>` adalah operator pengalihan output yang melanjutkan pengalihan data hingga nama file terakhir ditemukan, dan ini akan menjadi file tempat data akan disimpan. Lihat output yang diberikan di atas; file terakhir adalah `file3`, yang berisi 453 data ukuran, sedangkan ukuran file dan `file2` adalah 0. Berikut adalah panduan terperinci tentang berbagai parameter `gzip` dan kegunaannya. Kami juga dapat mengotomatiskan cadangan terkompresi menggunakan `crontab`. Mari kita pelajari itu di bagian berikut.

Gunakan `crontab` untuk Mengotomatiskan Pencadangan Terkompresi. Kami membuat tugas `cron` untuk mengotomatiskan prosedur pengambilan cadangan terkompresi dari database di mana tugas `cron` akan menjalankan `mysqldump` perintah pada waktu tertentu yang akan Anda tetapkan. Kita perlu mengikuti langkah-

langkah untuk mengatur backup terkompresi otomatis untuk database MySQL melalui cron job.

- Buat .my.cnf file di direktori home pengguna.

```
sudo nano ~/.my.cnf
```

Tulis teks yang tepat seperti yang diberikan di bawah ini.

```
[client]
```

```
user = db_username
```

```
password = db_password
```

Jangan lupa untuk memperbarui db_username dan db_password dengan kredensial Anda. Misalnya, nama pengguna adalah root, dan kata sandinya adalah 12345.

Contoh Kode:

```
[client]
```

```
user = root
```

```
password = 12345
```

Setelah selesai, simpan dan keluar dari file.

- : Batasi izin file kredensial. Dengan cara ini, pengguna yang ditentukan akan memiliki akses ke sana.

```
$ chmod 600 ~/.my.cnf
```

- : Selanjutnya, buat direktori untuk menyimpan semua cadangan di satu tempat.

```
$ mkdir ~/database_backups
```

- : Buka crontabfile sekarang.

```
$ crontab -e
```

Tulis pekerjaan berikut di crontabfile yang baru saja Anda buka. Ini akan membuat cadangan terkompresi basis data setiap hari pada pukul 3 pagi. Dalam pekerjaan berikut, nama pengguna adalah root, nama database adalah test, dan database_backupsfolder menyimpan semua cadangan.

```
$ 0 3* * * /usr/bin/mysqldump -u root test | gzip > /home/root/database_backups/test-$(date +%Y%m%d).gz
```

Mencadangkan Database Dengan Prosedur dan Fungsi

Service Set Identifier (SSID)

File cadangan yang dibuat oleh utilitas mysqldump pada dasarnya adalah seperangkat pernyataan SQL yang dapat digunakan untuk membuat ulang database asli. Perintah mysqldump juga dapat menghasilkan file dalam format CSV dan XML.

Anda juga dapat menggunakan utilitas mysqldump untuk mentransfer database MySQL Anda ke server MySQL lain.

Sintaks Perintah Mysqldump

Sebelum membahas cara menggunakan perintah mysqldump, mari kita mulai dengan meninjau sintaks dasar.

Ekspresi utilitas mysqldump mengambil bentuk berikut:

```
mysqldump > file.sql
```

```
options - File opsi mysqldump.sql - File dump  
(cadangan)
```

Untuk menggunakan perintah mysqldump, server MySQL harus dapat diakses dan dijalankan.

Cadangkan Database MySQL Tunggal

Kasus penggunaan paling umum dari alat mysqldump adalah untuk membuat cadangan satu database. Misalnya, untuk membuat cadangan dari database bernama `database_name` menggunakan `root` pengguna dan menyimpannya ke file bernama `database_name.sql` Anda akan menjalankan perintah berikut:

```
mysqldump -u root -p database_name > database_name.sql
```

Anda akan diminta memasukkan kata sandi `root`. Setelah otentikasi berhasil, proses dump akan dimulai. Bergantung pada ukuran database, prosesnya bisa memakan waktu.

```
mysqldump database_name > database_name.sql
```

Cadangkan Beberapa Database MySQL.

Untuk `--database -backup` beberapa database MySQL dengan satu perintah, Anda perlu menggunakan opsi `--database` diikuti oleh daftar database yang ingin Anda backup. Setiap nama basis data harus dipisahkan oleh ruang.

```
mysqldump -u root -p --databases database_name_a  
database_name_b > databases_a_b.sql
```

Perintah di atas akan membuat file dump yang berisi kedua database.

Cadangkan Semua Database MySQL.

Gunakan opsi `--all-databases` untuk mencadangkan semua database MySQL:

```
mysqldump -u root -p --all-databases >  
all_databases.sql
```

Sama seperti contoh sebelumnya, perintah di atas akan membuat file dump tunggal yang berisi semua basis data.

Cadangkan semua database MySQL ke file terpisah.

Utilitas `mysqldump` tidak memberikan opsi untuk mencadangkan semua basis data ke file yang terpisah, tetapi kami dengan mudah mencapainya dengan bash FOR loop sederhana:

```
for DB in $(mysql -e 'show databases' -s --skip-  
column-names); do mysqldump $DB > "$DB.sql"; done
```

Perintah di atas akan membuat file dump terpisah untuk setiap database menggunakan nama database sebagai nama file.

Buat Backup Database MySQL Terkompresi

Jika ukuran basis data sangat besar itu adalah ide yang bagus untuk mengompres output. Untuk melakukan itu cukup pipa output ke utilitas gzip , dan arahkan kembali ke file seperti yang ditunjukkan di bawah ini:

```
mysqldump database_name | gzip > database_name.sql.gz
```

Buat Cadangan dengan Stempel Waktu

```
mysqldump database_name > database_name-$(date +%Y%m%d).sql
```

Perintah di atas akan membuat file dengan format database_name-20180617.sql

Mengembalikan dump MySQL

Anda dapat memulihkan dump MySQL menggunakan alat mysql . Perintah sintaks umum adalah sebagai berikut:

```
mysql database_name < file.sql
```

Dalam kebanyakan kasus, Anda harus membuat basis data untuk diimpor. Jika database sudah ada, pertama Anda harus menghapusnya.

Pada contoh berikut ini, perintah pertama akan membuat database bernama database_name dan kemudian akan mengimpor dump database_name.sql ke dalamnya:

```
mysql -u root -p -e "create database database_name";  
mysql -u root -p database_name < database_name.sql
```


Kembalikan Database MySQL Tunggal dari Dump MySQL Penuh

```
mysql --one-database database_name < all_databases.sql
```

Ekspor dan Impor Database MySQL dalam Satu Perintah

Alih-alih membuat file dump dari satu database dan kemudian mengimpor cadangan ke database MySQL lain Anda bisa menggunakan satu-liner berikut:

```
mysqldump -u root -p database_name | mysql -h remote_host -u root -p remote_database_name
```

Perintah di atas akan menyalurkan output ke klien mysql pada host jarak jauh dan akan mengimpornya ke database bernama remote_database_name . Sebelum menjalankan perintah, pastikan database sudah ada di server jauh.

Otomatis Cadangan dengan Cron

Mengotomatiskan proses pencadangan basis data semudah membuat tugas cron apa yang akan menjalankan perintah mysqldump pada waktu yang ditentukan. Untuk mengatur cadangan otomatis dari database MySQL menggunakan cronjob, ikuti langkah-langkah di bawah ini:

Buat file bernama .my.cnf di direktori home pengguna Anda:

```
sudo nano ~/.my.cnf
```

Salin dan tempel teks berikut ini ke file.my.cnf.

```
user = dbuser password = dbpasswd
```

Jangan lupa ganti dbpasswd dan dbpasswd dengan pengguna basis data dan kata sandi pengguna. Batasi izin file kredensial sehingga hanya pengguna Anda yang dapat mengaksesnya:

```
chmod 600 ~/.my.cnf
```

Buat direktori untuk menyimpan cadangan:

```
mkdir ~/db_backups
```

Buka file crontab pengguna Anda:

```
crontab -e
```

Tambahkan tugas cron berikut yang akan membuat cadangan dari nama database mydb setiap hari pukul 3 pagi:

```
0 3 * * * /usr/bin/mysqldump -u dbuser mydb > /home/username/db_backups/mydb-$(date +%Y%m%d).sql
```

Jangan lupa mengganti username dengan username Anda yang sebenarnya. Anda juga dapat membuat cronjob lain untuk menghapus cadangan yang lebih dari 30 hari:

```
find /path/to/backups -type f -name "*.sql" -mtime +30 -delete
```

Tentu saja, Anda perlu menyesuaikan perintah sesuai dengan lokasi cadangan Anda dan nama file. Untuk mempelajari lebih lanjut tentang perintah find, periksa Cara Menemukan File di Linux Menggunakan panduan Command Line.