



Nurul Huda
18 January 2020

Beranda > Database > MySQL

Backup MySQL Secara Otomatis Dengan Cron Job

Backup **MySQL** secara
Otomatis dengan
Cron Job

Daftar Isi

Pendahuluan

Sekilas Tentang Cron Job

Konfigurasi Cron Job dengan crontab

Atur hasil *backup* dengan format tanggal dan jam yang jelas

Backup Perjam

Backup Sekali Sehari

Kompres Hasil Backup dengan Gzip

Hapus file Backup yang lebih dari 10 hari

Kesimpulan

Bagikan:



Pendahuluan

MySQL atau pun MariaDB adalah salah satu database yang paling banyak digunakan. Apalagi jika menggunakan jasa *shared hosting* di Indonesia, kebanyakan pasti menggunakan MySQL mau pun MariaDB secara default.

Ketika situs/maupun aplikasi web kita telah tumbuh semakin besar, data yang tersimpan dalam database menjadi suatu asset yang sangat berharga. Wajib kita lakukan *backup* secara berkala agar jika terjadi sesuatu, kita tidak menyesal di kemudian hari. Betapa banyak orang yang kebingungan karena database-nya korup, tidak sengaja terhapus, atau di-*hack* orang dan mereka tidak memiliki *backup*-nya?

CPanel, kalian tetap bisa mengikuti artikel ini karena di CPanel pun terdapat menu pengaturan cron job.

Baca juga: [MySQL vs MariaDB. Mana yang harus dipilih?](#)

Sekilas Tentang Cron Job

Cron Job adalah aplikasi atau layanan yang memungkinkan kita untuk melakukan suatu tugas tertentu secara terjadwal tanpa harus melakukannya secara manual. Misal: saya ingin melakukan duplikat data dari situs A ke situs B setiap jam 1 malam. Maka saya bisa menggunakan cron job.

Cron job bisa diinstruksikan untuk mengerjakan satu tugas tertentu satu kali saja di waktu yang ditentukan, atau ia juga bisa dikonfigurasi untuk melakukan pekerjaan yang sifatnya repetitif atau berulang.

```
# |----- menit (0 - 59)
# | |----- jam (0 - 23)
# | | |----- tanggal (1 - 31)
# | | | |----- bulan (1 - 12)
# | | | | |----- hari (0 - 6)
# | | | | | (Dari Minggu ke Sabtu; Angka 7 juga
# | | | | | dianggap hari Minggu di beberapa sistem)
# | | | | |
# * * * * * [perintah yang akan dieksekusi]
```

Ilustrasi 1: pengaturan dan konfigurasi cron job dengan aplikasi `crontab`

Misalkan saya ingin mengeksekusi perintah `ping google.com` setiap 5 menit, saya akan menambahkan kode berikut di baris paling bawah konfigurasi cron job:

Konfigurasi Cron Job dengan crontab

Untuk mengkonfigurasi Cron Job di linux, kita perlu menggunakan aplikasi bernama `crontab`. Jika anda belum menginstall `cron` atau pun `crontab`, anda bisa menginstallnya terlebih dahulu.

Ubuntu

Untuk pengguna ubuntu, kita bisa langsung install `cron` melalui `apt`.

```
sudo apt update  
sudo apt install cron
```

Arch Linux

Di arch linux, terdapat beberapa aplikasi yang mengimplementasikan fungsi `cron`, di antaranya adalah: `crone` dan `fcron`. Anda cukup memasang salah satunya saja.

```
sudo pacman -S crone
```

Atur hasil *backup* dengan format tanggal dan jam yang jelas

Kita akan menambahkan *postfix* tanggal dan waktu pada setiap file `.sql` dari hasil backup yang kita lakukan. Kita bisa mendapatkan informasi tanggal dan jam pada

Berikut ini adalah perintah untuk mendapatkan tanggal dan jam pada linux dengan format `tanggal-bulan-hari@jam:menit`.

```
date '+%Y-%m-%d@%H:%M'
```

Perintah di atas (ketika saya eksekusi) menghasilkan output berikut:

```
2020-01-25@21:52
```

Output tersebut sesuai dengan waktu ketika perintah `date` dieksekusi. *Well*, karakter `:` tidaklah dilarang di Linux. Itu artinya kita bisa menggunakan karakter tersebut baik untuk nama file mau pun untuk nama direktori di linux.

Untuk perintah lengkap mem-*backup* mysql adalah sebagai berikut:

```
mysqldump -u [username] -p[password] [nama-db] > [output-db]
```

Katakanlah saya mempunyai variabel:

- username: `root`
- password: `12345`
- nama db: `kampus_db`
- nama output file: `kampus_db_`date '+%Y-%m-%d@%H:%M'`.sql`

Maka perintah akhir yang saya eksekusi adalah:

```
mysqldump -u root -p12345 kampus_db > kampus_db_`date '+%Y-%m-%d@%H:%M'`.sql
```

Backup Perjam

Di atas kita telah membuat perintah untuk membackup mysql dengan output nama file yang berisi tanggal dan waktu *backup*. Sekarang kita akan mengkonfigurasi `cronjob` untuk mengeksekusi perintah di atas setiap 1 jam sekali.

Edit `crontab` dengan melakukan perintah:

```
crontab -e
```

Lalu pada baris paling bawah, tambahkan perintah berikut:

```
0 * * * * mysqldump -u root -p12345 kampus_db > kampus_db_`date '+%Y-%m-%d@%H:%M'`
```

Statement di atas akan menginstruksi `cron` untuk membackup database `kampus_db` setiap menit ke-0 untuk setiap jam, untuk setiap hari. Alias ia akan mengeksekusi perintah tersebut setiap 1 jam sekali.

Backup Sekali Sehari

Untuk membackup sekali sehari pada pukul tertentu, kita hanya perlu mengisi bagian `menit` dan bagian `jam`. Sisanya kita isi dengan `*`.

Misalkan setiap jam 00:00

Untuk setiap jam `00:00`, saya atur `menit` dan `jam`-nya menjadi angka `0`:

Misalkan setiap jam 2 pagi

Untuk setiap jam `02:00` pagi, saya atur `menit`-nya dengan angka `0`, dan `jam`-nya menjadi angka `2`:

```
0 2 * * * mysqldump -u root -p12345 kampus_db > kampus_db_`date +%Y-%m-%d@%H:%M``
```

Untuk memudahkan konfigurasi waktu, anda bisa mengunjungi crontab.guru. Di sana anda bisa mensimulasikan konfigurasi waktu `crontab` langsung via web.

Kompres Hasil Backup dengan Gzip

File hasil *backup* mysql ukurannya bergantung pada ukuran data kita dalam database. Semakin banyak baris data di dalam database kita, maka hasil *backup*-nya pun juga akan semakin besar. Untuk mensiasati hal tersebut, kita bisa mengkompresnya menggunakan software `Gzip`.

Anda bisa mengganti perintah *backup* mysql yang sudah kita lakukan di atas menjadi seperti ini:

```
mysqldump -u root -p12345 kampus_db | gzip > kampus_db_`date +%Y-%m-%d@%H:%M`.sc
```

Baca juga: [Cara Kompres dan Ekstrak File .gz Di Linux](#)

Hapus file Backup yang lebih dari 10 hari



3GB . Lumayan bukan? Bisa menghabiskan **storage** kita untuk hal yang tidak terlalu dibutuhkan.

Untuk menghindari hal tersebut, kita bisa melakukan proses penghapusan data *backup* yang lebih dari 10 hari terakhir. Karena toh kita kemungkinan besar hanya membutuhkan data yang paling baru, data yang sudah lama sekali hanya akan memenuhi kapasitas storage jika terus kita simpan.

Berikut adalah langkah-langkah menghapus file *backup* yang sudah lebih dari 10 hari.

1. List file *backup* terlebih dahulu

```
ls
```

Contoh output milik saya. Terdapat 15 file dari tanggal 1 Januari 2020 sampai tanggal 15 Januari 2020. Nantinya kita hanya akan mengambil yang lebih dari 10 file terakhir.

```
kampus_db_2020-01-01@09:14.sql  kampus_db_2020-01-09@09:06.sql
kampus_db_2020-01-02@09:14.sql  kampus_db_2020-01-10@09:06.sql
kampus_db_2020-01-03@09:14.sql  kampus_db_2020-01-11@08:54.sql
kampus_db_2020-01-04@09:14.sql  kampus_db_2020-01-12@08:54.sql
kampus_db_2020-01-05@09:06.sql  kampus_db_2020-01-13@08:54.sql
kampus_db_2020-01-06@09:06.sql  kampus_db_2020-01-14@08:54.sql
kampus_db_2020-01-07@09:14.sql  kampus_db_2020-01-15@08:54.sql
kampus_db_2020-01-08@01:06.sql
```

2. Urutkan file secara **descending** berdasarkan nama

membalik hasil urutan, alias mengurutkannya dari yang paling belakang atau *descending*.

```
ls | sort -dr
```

Output:

```
kampus_db_2020-01-15@08:54.sql
kampus_db_2020-01-14@08:54.sql
kampus_db_2020-01-13@08:54.sql
kampus_db_2020-01-12@08:54.sql
kampus_db_2020-01-11@08:54.sql
kampus_db_2020-01-10@09:06.sql
kampus_db_2020-01-09@09:06.sql
kampus_db_2020-01-08@01:06.sql
kampus_db_2020-01-07@09:14.sql
kampus_db_2020-01-06@09:06.sql
kampus_db_2020-01-05@09:06.sql
kampus_db_2020-01-04@09:14.sql
kampus_db_2020-01-03@09:14.sql
kampus_db_2020-01-02@09:14.sql
kampus_db_2020-01-01@09:14.sql
```

3. Skip file 10 file yang pertama

Kita akan menampilkan file yang lebih dari 10 hari dengan menskip 10 baris pertama dari hasil output perintah sebelumnya. Kita tambahkan perintah `tail -n +11`.

```
ls | sort -dr | tail -n +11
```

```
kampus_db_2020-01-05@09:06.sql  
kampus_db_2020-01-04@09:14.sql  
kampus_db_2020-01-03@09:14.sql  
kampus_db_2020-01-02@09:14.sql  
kampus_db_2020-01-01@09:14.sql
```

Sekarang kita hanya memiliki 5 file saja dari tanggal 5 Januari 2020 sampai tanggal 1 Januari 2020.

4. Hapus file yang lebih dari 10 hari

Untuk menghapus file yang lebih dari 10 hari. Kita tambahkan `xargs rm` pada perintah sebelumnya setelah menambahkan tanda `| pipe`.

```
ls | sort -dr | tail -n +11 | xargs rm
```

5. Hapus file secara rutin

Untuk menghapus file secara rutin, tambahkan perintah di atas di dalam `crontab`.

```
0 0 * * * ls | sort -dr | tail -n +11 | xargs rm
```

Kesimpulan

Database adalah salah satu aset paling mahal untuk suatu perusahaan. Maka dari itu wajib untuk di-*backup* secara berkala. Dengan memanfaatkan `cronjob`, kita bisa memerintahkan sistem untuk membackup database MySQL sesuai kebutuhan kita: bisa sehari sekali, atau setiap jam sekali, atau setiap hari senin, atau setiap tanggal 1 dan tanggal 10 tiap bulannya, dan sebagainya.

server kita tidak membengkak hanya gara-gara file *backup* yang terlalu banyak.

Bagikan:



Nurul Huda

Web Developer. FOSS addict. Pengguna Arch Linux (dan Ubuntu). Penyuka kopi saset. Dan pernah kuliah Teknik Informatika sampai lulus.

Dukung Jago Ngoding 

Artikel Terkait



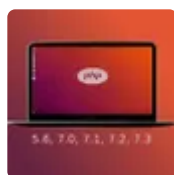
Import Dan Export MySQL



5 Aplikasi Alternatif phpMyAdmin



MariaDB vs MySQL. Mana Yang Harus Dipilih?




Cara Install Lebih Dari Satu Versi PHP (5.6, 7.x) di Ubuntu

0 Comments - powered by utteranc.es

Write

Preview

Sign in to comment

 Styling with Markdown is supported

Sign in with GitHub



© 2022 Jago Ngoding

Icons made by Freepik from www.flaticon.com