

How to schedule a MySQL database backup on Ubuntu

Published on: 18 January 2020

Backup

Database

MySQL

Contents

[How to schedule a MySQL database backup on Ubuntu](#)

- Use mysqldump
- Automate mysqldump through crontab
- Use automysqlbackup
- Customizing automysqlbackup

Performing regular backups is an essential procedure for any type of data. This tutorial will help you learn about some of the most used tools to backup and restore your MySQL database.

First you will need to connect to your server via an SSH connection. If you have not already done so, we recommend that you follow our guide to [connect securely with SSH](#). In case of local server you can go to the next point and open the terminal of your server.

Use mysqldump

Mysqldump is an application always available in every MySQL installation and allows you to perform a complete dump of your data in text format.

The command syntax is as follows:

```
$ mysqldump -u [username] -p [database] > backup.sql
```

By running this command, you will find a complete database backup indicated in the backup.sql file.

To restore the backup use the MySQL client as follows:

```
$ mysql -u [username] -p [database] < backup.sql
```

Depending on the size of the database, these backup files can reach significant sizes, making it difficult to transfer or copy them. To optimize the size of the database backup, you can use gzip to compress the file:

```
$ mysqldump -u [username] -p [database] | gzip > backup.sql.gz
```

If you wish, you can also automatically add the current date in the filename using:

```
$ mysqldump -u [username] -p [database] | gzip > backup_$(date +%F.%H%M%S).sql.gz
```

To restore, use the following syntax instead:

```
$ mysql -u [username] -p [database] < backup.sql
```

Notes on mysqldump:

It is good to keep in mind that during the dump and backup of a database, mysqldump does not guarantee the consistency of data between tables. If write operations occur during the backup, the backup obtained may be partial and / or corrupt.

To prevent this problem, there are specific options provided by the application:

- lock-all-tables: blocks all write operations on all tables of all databases
- lock-tables: this option instead sets the database table being backed up in read-only mode.

Which options to choose depends on whether you have data related to multiple databases or not. In most cases --lock-tables is the ideal solution to guarantee data integrity for each individual database.

Automate mysqldump through crontab

To make the backup periodic, all you have to do is add it to the system's crontab by programming, for example, once a day. First, open the crontab by starting:

```
$ sudo crontab -e
```

Now add a line at the end of the file as follows:

```
00 23 * * * mysqldump -u [username] -p [database] | gzip > /home/utente/backup.sql.gz
```

By saving the file, the system will make one backup per day (at 11:00 pm) within your user directory, overwriting the previous backup every day.

Use automysqlbackup

In addition to using mysqldump by exporting backups manually, you could rely on one of the most popular utilities in the industry: automysqlbackup.

First you need to install the package through apt-get:

```
$ sudo apt-get install automysqlbackup
```

After completing the installation, automysqlbackup will perform daily backups in the /var / lib / automysqlbackup directory. You can also manually start a backup by launching:

```
$ sudo automysqlbackup
```

Once completed, you can verify the present backups simply by listing the contents of the directory:

```
$ ls -R /var/lib/automysqlbackup/

/var/lib/automysqlbackup/:

daily  monthly  weekly
/var/lib/automysqlbackup/daily:

exampledb  information_schema  performance_schema  test  wordpress
/var/lib/automysqlbackup/daily/exampledb:

exampledb_2019-08-03_06h48m.Sunday.sql.gz

...
```

If everything worked properly, you'll notice a series of files with .sql.gz extensions as shown above.

We advise you to periodically check the latest backups made and verify that they all work properly.

Customizing automysqlbackup

To change the backup frequency of automysqlbackup or the access data needed to access your database you can modify the configuration file in /etc/default/automysqlbackup.

Some of the settings are:

- USERNAME: username used to log in to the db
- PASSWORD: database password
- DBHOST: IP address or host name
- DBNAMES: names of the databases to be backed up
- BACKUPDIR: Directory to save backups

After changing the desired settings, save the file to apply the changes.

A complete Cloud environment for developing your projects



Cloud PRO
Learn more >



Cloud VPS
Learn more >



Private Cloud
Learn more >



Cloud Backup
Learn more >



Database as a Service
Learn more >



Cloud Object Storage
Learn more >



Domain Center
Learn more >



Jelastic Cloud
Learn more >

TECHNOLOGICAL PARTNERS :



- Support
- Report abuse
- Login
- Top-up credit
- Use your voucher

PRODUCTS & SOLUTIONS

Cloud VPS
Cloud PRO
Jelastic Cloud
Database as a Service
Private Cloud
Cloud Backup
Cloud Object Storage
Domain Center
Cloud Monitoring

COMPANY

About us
Why Choose Aruba
Aruba Racing Team
News
Partners
Data protection in the EU

NETWORK

Aruba Cloud Italy
Aruba Cloud International
Aruba Cloud France
Aruba Cloud Spain
Aruba Cloud Hungary
Aruba Cloud Poland

