

BEATIFICA BYTES

The Joys of Computing

Schedule a Backup of all your MySQL databases on Synology

Posted on [September 29, 2012](#)

It would be a good idea to schedule a daily backup of all the MySQL databases on your Synology. Those will indeed be wiped out if by accident, you have to reset your NAS (reinstall the firmware), ex.: if you lose a disk in your RAID 0.

That can be done using a shell script and a cron job as described here after.

Click to close

First, create a user with a local-read-only access on the databases. We will indeed have to let its password in the script, endangering the security. Using a local-read-only user will mitigate the risks.

1. Go to the web administration interface (DSM) of your NAS.
2. Install phpMYAdmin via the "Package Center" if not yet done.
3. Open phpMyAdmin via the "Start Menu" ("Main Menu").
 - A. I am using the login 'root' with the password of Syno's 'admin' user
4. Open the "Users" tab in phpMyAdmin
5. Click "Add User" (bellow the list of existing users)

6. Enter the User Name. Ex.: backup
7. Select "localhost" as an Host.
8. Enter a password. Ex.: mysql
9. Keep "none" as "Database for user"
10. Grant the user with Global privileges: Select (in Data) and Reload, Show Databases and Lock Tables (in Administration)
11. Click "Add User" at the botton of the page

Next, create a shell scripts in a Shared Folder of the NAS (Ex.: \volume1\backup\backupMySql.sh). Working in a Shared Folder will make it easier for you to copy/open later the backups from your PC). Don't forget to create a "Unix" file, either using the touch command in a Console or saving a file As a "Unix shell script" within Notepad++. Notice that a script created with NotePad++ and saved on the Shared Folder will belong to the user account accessing that Shared Folder (Most probably your Windows Account if like me you simply created a user on your NAS with the same login and password). A script created with "touch" in a Console will belong to the user accessing the NAS via telnet/SSH (Most probably the "root" account).

```

1  #!/bin/sh
2  #
3  DIR=/volume1/backup/sqlbackup/
4  DATESTAMP=$(date +%Y%m%d%H%M%S)
5  DB_USER=backup
6  DB_PASS=mysql
7
8  # create backup dir if it does not exist
9  mkdir -p ${DIR}
10
11 # remove backups older than $DAYS_KEEP
12 #DAYS_KEEP=30
13 #find ${DIR}* -mtime +$DAYS_KEEP -exec rm -f {} \; 2>&t; /dev/null
14
15 # remove all backups except the $KEEP latest
16 KEEP=5
17 BACKUPS=`find ${DIR} -name "mysqldump-*.gz" | wc -l | sed 's/\ //g'`
18 while [ $BACKUPS -ge $KEEP ]
19 do
20     ls -tr1 ${DIR}mysqldump-*.gz | head -n 1 | xargs rm -f
21     BACKUPS=`expr $BACKUPS - 1`
22 done
23
24 #
25 # create backups securely
26 #umask 006
27
28 # dump all the databases in a gzip file

```

```
29 | FILENAME=${DIR}mysqldump-${DATESTAMP}.gz
30 | /usr/syno/mysql/bin/mysqldump --user=$DB_USER --password=$DB_PASS --opt --all-databases --flush-logs | g:
```

NB: Since DSM 6.0, "/usr/syno/mysql/bin/mysqldump" has been moved to "/bin/mysqldump" !!!

Possibly, type the following command in a Console (telnet/SSH) to set the user 'root' as owner of the script:

```
1 | chown root /volume1/backup/backupMySQL.sh
```

IMPORTANT notice: I used to have "-u DB_USER" (with a blank space in between) in my script above and "-p\$DB_USER" (without a blank) instead of --user and --password. But a reader (Fredy) reported that the script was not running fine for him unless removing the blank. As per documentation of mysqldump's [user parameter](#) and [password parameter](#), there should be a blank after -u but not after -p. However, samples on the web usually illustrate the use of mysqldump with "-uroot". So, I decided to use the more explicit notation "--user=" and "--password=". I did test this notation with a wrong username or password and the resulting dump is indeed empty. With the correct username and password, it works fine.

Since DSM 4.2, the Task Scheduler can be used to run the script on a daily basis.

1. Go to the Start Menu
2. Open the Control Panel
3. In the "Application Settings", open the Task Scheduler
4. Select "Create a User-Defined Script"
5. Type a name for that "Task:"
6. Keep the "User:" root
7. In the "Run Command" pane, type:

```
sh /volume1/backup/backupMySQL.sh
```

Don't forget the "sh" in front of your command, otherwise, it does not work (although the sample provided by Synology is missing it ?!)

If you don't use the Task Scheduler, you can add a Cron Job to execute the script, e.g. every day at 0:01. Open a Console (Telnet/SSH) and type:

```
1 | echo "1 * * * * root sh /volume1/backup/backupMySQL.sh" && /etc/cr
```

FYI, a cron line syntax is "mm hh dd MMM DDD user task" where:

- mm is the minute (0..59)
- hh is the hour (0..23)
- dd is the day in the month (1..31)
- MMM is the month (jan, feb, ... or 1..12)
- DDD is the day in the week (sun, mon, ... or 0..7 where 0 and 7 beeing both sunday)

The following values can also be used:

- * : a every unit (0, 1, 2, 3, 4...)
- 5,8 : at unit 5 et 8
- 2-5 : units 2 to 5 (2, 3, 4, 5)
- */3 : every 3 units (0, 3, 6, 9...)
- 10-20/3 : every 3 units, from 10th to 20th (10, 13, 16, 19)

So, the script will start every day at 0:01h

Finally, you must restart the cron daemon to activate the new job. To find the process id and send a SIGHUP signal to this one, type the following command line in a Console (Telnet/SSH):

```
1 | ps | grep crond | grep -v grep | awk '{print$1}' | xargs -t kill -HUP
```

It should display "Kill -HUP xxxx" where xxx is the pid of the cron daemon.

Click to Close

Attention: if you copy paste the command to restart the cron daemon from this page into the telnet console, some symbols will be wiped out: the `{` and ` ` around print\$1... you have to type them manually...

Attention: if you upgrade the version of DSM, the changes made in the cron script are lost (at least, it's what I have noticed after updating last month...). Reason why I recommend to use the new "Task Scheduler" feature available since DSM 4.2.

Tip to edit the cron script: you can use the package "Config File Editor" available [here](#). Once installed, you can access it via the Main Menu. Then, you have to edit the file named "Config File Editor" and add the following line:

```
/etc/crontab,crontab
```

Once this line is added and saved, ... I don't remember how to force the change to be taken into account :/. But restarting the DiskStation is enough :D

Finally, back into the Config File Editor, select the 'crontab' entry and modify this script, save your changes and restart the cron daemon.

Tip to restart the cron daemon: you can use the package "Web Console" available [here](#). To install it, go to the Package Center and add the following url via the Settings > Package Sources : <http://missilehugger.com/708/synology-package-web-console>. Once this url added, go to the tab "Other Sources" and click Install on the "Web Console" icon.

Run the Web Console via the Main Menu, log in with "admin"/"admin" (The defaults values if not changed) and type:

```
synoservice --restart crond
```

NB: Since DSM 6.0, "Web Console" does not work anymore.

Click to Close

If you want a lazy solution to notify the Administrator about the success or failure of the backup, you can use the 'synonotify' command (See [details here](#)). A more advanced solution would be to configure the "Mail Server" and use its 'sendmail' command: /volume1/@appstore/MailServer/sbin/sendmail...

Click to Close

mysqldump has moved

- DSM 6.0 : /usr/syno/mysql/bin/mysqldump
- MariaDB 5: /volume1/@appstore/MariaDB/usr/bin/mysqldump
- MariaDB 10: /volume1/@appstore/MariaDB10/usr/local/mariadb10/bin/mysqldump

▪

Click to Close

To do a full restore, simply:

1. Copy the archive (.gz) you plan to restore on your local PC
2. Go to "phpMyAdmin" > "Import" > "Chose File" (next to *Browse your computer*).
3. Select the .gz file to be restored
4. Press "Go" at the bottom of the page (no need to change any settings)

Pay attention: this is a complete restore, I.e.: all databases are dropped and recreated from scratch.

If you want to be able to restore a subset of your databases you better have to change the script to backup specific databases instead of --all-database.

If you want to restore only one table 'mytable' from the backup '*mysqldump-datetime.gz*':

1. Unzip first the archive: `gunzip mysqldump-datetime.gz`
2. Then extract the part that restore the desired table with this command: `sed -n -e '/DROP TABLE.*mytable/,/UNLOCK TABLES;/p' mysqldump-datetime > mytabledump`

Download the script [here](#).



This entry was posted in [Synology](#) and tagged [Backup](#), [DS209+](#), [MySQL](#) by [Valery Letroye](#). Bookmark the [permalink](#) [<http://www.BeatificaBytes.be/schedule-a-backup-of-all-your-mysql-databases-on-synology/>]

50 THOUGHTS ON "SCHEDULE A BACKUP OF ALL YOUR MYSQL DATABASES ON SYNOLOGY"



Valery Letroye

on **March 3, 2013 at 23:22** said:

Instead of scheduling the backup manually in the crontab file, a task can be created via the new Task Scheduler of DSM 4.2... It's far easier as this tool restart automatically the cron daemon :)

I have updated my post with this info.

Notice however that I had to type explicitly "sh" in front of the script to make it work as a Scheduled Task...

Also, I am now using "Time Backup" to backup "/volume1/backup/sqlbackup/" every day onto a USB key which stays plugged 24/7 on the Synology. Works like a charm too.



Paul

on **April 16, 2013 at 16:31** said:

Thanks for posting this. Unfortunately when I tried to run the task it created the sqlbackup folder but no backup files. Just a folder icon with no name and no files within it. Not sure what I'm doing wrong. Using DSM v4.2.

**Valery Letroye**

on **April 16, 2013 at 21:41** said:

Do you use the new Task Scheduler to run the script ?

Possibly try to run the script manually for diagnostic purpose:

- Open a cmd prompt on your PC and execute:telnet your_syno_address
- login as "root" with your admin password and execute: sh /volume1/backup/backupMySQL.sh

Do you see any dump now in the output folder or any error displayed by the shell ?



Paul

on **April 16, 2013 at 22:15** said:

It reports not found backup's for lines 7,10 and 14, but those are the spaces in your script above, so shouldn't be a problem. Then says:

```
"No such file or directory //volume1/backup/backupMySQL/.sh: Line 30 : syntax error : unexpected end of file (expecting "do")"
```



Paul

on **April 16, 2013 at 23:47** said:

OK, I tried vi backupMySQL.sh and there were a lot of bogus ^M's at the end of each line. So I think there must be a translation error when copying in the script. Having removed those and re-running the script it does create a mysqldump.gz file but only of 20 bytes. The shell says:

```
"error 1045: Access denied for user backup@localhost (using password : YES) when trying to connect"
```



Valery Letroye

on **July 19, 2013 at 13:04** said:

Check that you selected "Local" as an Host for your 'backup' user. In phpMyAdmin, you should see 'localhost' in the column 'Client' for this user. If you did, try to reset/change the password of this user.



Paul

on **April 17, 2013 at 00:27** said:

Well it appears somehow that the user I had created in phpmyadmin got deleted. When I recreated I then got a show view error, so I added that privilege at your step 10 and finally I got it to work with no errors.

Maybe at least my exploits will assist anyone equally boneheaded to get the script to work.

Thanks Valery



Valery Letroye

on **April 24, 2013 at 09:39** said:

Thanks a lot for posting your findings! It can indeed help other readers.



Sylvain

on **April 26, 2013 at 11:14** said:

Great little script ! Work like a charm on the first test.



Ulrich

on **July 17, 2013 at 11:20** said:

thanks for your script, but i have still problems...
telnet: no such file or directory
after start the script, the new folder sql is added, but its empty

is there a solution?

thanks
ulrich



Valery Letroye

on **July 19, 2013 at 12:52** said:

Did you read the remark from Paul. He had the same issue and this was due to the Copy/Paste of the script from this blog. Be sure that you use notepad++ to paste the script into a file created with the "touch" command. Doing so, the encoding should be correct.



Per-Åke Franklind

on **July 25, 2013 at 19:03** said:

It works perfect, just one question.

How do you backup images? I did a test to re-install and it worked except for broken links for images.



Valery Letroye

on **July 26, 2013 at 00:41** said:

Actually, when I have new pictures, I copy them from my PC on the NAS and next on my backup server (which is not on 24/7. I turn it only on purpose).

But I could schedule a backup directly from the NAS on the backup server using the native backup/restore feature of Synology - either with an rsync service installed on my server or with the Sync Shared Folder feature. In both cases, you can select the various shared folders to be backedup, e.g. video, photo, ...

**Per-Åke Franklind**

on **July 26, 2013 at 08:26** said:

Hmmm but it looks like you also are using the upload function in Media Library. Image url for your images are like /wordpress/wp-content/uploads/2013/07/xxx.jpg

So even if I put all images in a folder like /volume1/web/image/ before uploading I still have the problem with broken URL links after a fresh installation, or have I misunderstood something (still a beginner)

**Valery Letroye**

on **July 27, 2013 at 12:14** said:

Sorry, I didn't understand that you were talking about those pictures in particular. In my case they are backedup as part of the "web" shared folder of Synology. I.e.: With all customisations done on WordPress style sheets, etc... (WordPress is a sub-folder of the "Web" folder)

So, I don't understand why you have broken links after restoring the shared folder "web"... I admit I didn't try to restore yet. As I recently discovered how to setup DSM in a virtual machine, I could do that to test a restore of a full backup.



Per-Åke Franklind

on **July 29, 2013 at 07:19** said:

Y, this folder and sub folders must be incl. in backup

/Volumes/web/wordpress/wp-content/uploads

I'm about to test BackUpWordPress plugin, this plugin backup the complete wordpress folder and/or database to a zip file.

- 1) save X no of backup in another folder on Synology (for easy access)
- 2) backup this folder to my external WD notebook



ward

on **September 14, 2013 at 23:45** said:

Worked like a charm. Thanks!

I have a virtual webserver next to my Synology NAS. The VDI is backed up to the NAS on a daily basis. The www dir is mounted to the NAS with NFS and now I was able to put the database on the NAS also. This will give me full backup ability. Next step will be a backup over the internet to a remote NAS



Gary

on **October 21, 2013 at 20:18** said:

Thanks for the script and detailed instructions! It worked like a charm on my DSM4.3 system. Cheers!



Guido

on **November 30, 2013 at 22:31** said:

It works really well on my DS213. I'm not good in Linux so I would like to ask you, is it possible to send een e-mail if backup is oke or not.

Thank you



Valery Letroye

on **December 1, 2013 at 00:23** said:

It's possible and there are several options: using the command 'sendmail' (/volume1/@appstore/MailServer/sbin/sendmail) and the 'Mail Server' of Synology to be installed via the Package Center, using a client mail to be installed with ipkg, etc ...

But if you are not comfortable with Linux, I would suggest you to use the same tool as Synology to send notifications. It's great IMO as you will get both an email AND a notification popup in your DiskStation UI.

First, be sure that you have configured your Synology to send notifications: "Control Panel" > "Notification" > "General\E-Mail" > "Enable e-mail notifications".

Then in the backup script, depending on the result of the dump, you can use:
`/usr/syno/bin/synonotify DatabaseBackupFailed`

```
/usr/syno/bin/synonotify DatabaseBackupCompleted
```

In a shell, the variable "\$?" equals "0" if the last command was successful. So, your script could end with something like this:

```
if [ $? -eq 0 ]  
then /usr/syno/bin/synonotify DatabaseBackupCompleted  
else /usr/syno/bin/synonotify DatabaseBackupFailed  
fi
```

'synonotify' is the tool of Synology I have mentioned.

'DatabaseBackupFailed' and 'DatabaseBackupFinished' are two 'tags' that you must define in the file /usr/syno/synoman/webman/texts/enu/mails (you possibly have to replace 'enu' by your own language: 'nld' I presume ?)

For each of those two new tags, you have to follow the syntax that you can see for existing tags in that file. I.e.: each tag must be between brackets and followed by something like:

- a line with: Category: Backup/Restore
- a line with: Title: Database backup
- a line with: Subject: Database backup on %HOSTNAME% is successfull
- and finally some text...

As soon as I have some free time, I will implement that for myself and write a post with the details. All info here above is 'theoretical'.



Guido

on **December 1, 2013 at 13:26** said:

Thanks for the answer. I will try to sort this out with the explanation above.
(nld is correct!)

Guido



Valery Letroye

on **December 1, 2013 at 22:41** said:

More details on how to proceed here: <http://www.beatificabytes.be/wordpress/send-custom-email-notifications-from-scripts-running-on-a-synology/>



Fredy

on **January 17, 2014 at 11:57** said:

Hi

I do have the same problem as Paul and Ulrich.

Mei GZ file extracted is size 0.

I did create the file witch "touch" and edited in Notepad++ - no change.

could you please help.

Thank you.



Fredy

on **January 17, 2014 at 13:39** said:

Got it

Problem seemed to be the -u \$DB_USER

Should be -u\$DB_USER with no space between -u and \$DB_USER



Valery Letroye

on **January 17, 2014 at 21:58** said:

Really weird... I just checked my own script which is running fine and export all my DB's every day, and there is a blank space between -u and \$DB_USER... But if I remove the blank space, the script still run fines too.

As per documentation of "mysqldump", there should be a blank space? So I did update my script to use the notation --user=... and --password=... which is not as ambiguous.

1000 thx for reporting your finding!



jap2slow

on **January 29, 2014 at 21:32** said:

nice work!!

how can i set it up to do only some databases?

i dont what all my sql, only 2



Valery Letroye

on **January 29, 2014 at 22:30** said:

You should backup each database and add the backups into a "tar" before compressing this "tar" into a "gz".
But I presume that you are asking as you are not familiar with Linux ?!

So, the easiest will be to backup each database in a separated "gz".

Remove the line with "/usr/syno/mysql/bin/mysqldump" and replace it with:

```
/usr/syno/mysql/bin/mysqldump --user=$DB_USER --password=$DB_PASS --opt --database your_database1 --flush-logs | gzip > ${DIR}mysqldump-${DATESTAMP}.database1.gz  
/usr/syno/mysql/bin/mysqldump --user=$DB_USER --password=$DB_PASS --opt --database your_database2 --flush-logs | gzip >> ${DIR}mysqldump-${DATESTAMP}.database2.gz
```

As you will have two backups per day, you will also have to replace "KEEP=5" with "KEEP=10"



Matthias Fröb

on **February 23, 2014 at 17:57** said:

Thx a lot. Your detailed instructions helped me very much.



Matthias Fröb

on **February 23, 2014 at 17:58** said:

How can I test to restore the database?



Valery Letroye

on **February 25, 2014 at 22:50** said:

Hi,

I did add a paragraph at the end of the post to explain how to restore...

Hope it helps !



Matthias Fröb

on **April 6, 2014 at 17:29** said:

thx a lot for Your very usefull instructions



Matthias Fröb

on **April 6, 2014 at 18:10** said:

I have got an further question: Can I encrypt the backup or at least secure it with a password? How would that alter the restore process?



Valery Letroye

on **April 6, 2014 at 19:41** said:

Unfortunately gzip does not support password encryption.

The safest solution would be to insert an encryption command (based on public/private certificates) between sqldump and gzip.

But if its ok for you, I would simply suggest to add 2 extra steps to encrypt with the 'zip' command and delete the original file.

```
/usr/syno/mysql/bin/mysqldump --user=$DB_USER --password=$DB_PASS --opt --all-databases --flush-logs | gzip > $FILENAME  
zip -j --password $FILENAME.zip $FILENAME  
rm $FILENAME
```

It's not great as your password is visible in the script. but at least, you need nothing more than your password to extract the dump (.gz) with any standard unzip tool installed on your PC.



Wall

on **May 3, 2014 at 16:04** said:

Thanks for the script works great ;)

Do you have one for backup mails, i'm running an mail server on my Synology nas.

**Valery Letroye**

on **May 3, 2014 at 16:38** said:

Unfortunately no, as I am not (yet) using the mail server...



Patrick

on **June 9, 2014 at 21:07** said:

I followd the instructions above, but I receive a syntax error:

```
syntax error: unexpected end of file (expecting ")")
```

Line 29:

```
/usr/syno/mysql/bin/mysqldump --user=$DB_USER --password=$DB_PASS --opt --all-databases --flush-logs |  
gzip > $FILENAME
```

What can be the error?



Valery Letroye

on **June 9, 2014 at 21:22** said:

Possibly an issue when copying text from my blog into your text file... WordPress is displaying double dash as a long dash (--), it transforms quotes, etc... And so, it results in errors like yours.

To make it easy, here is a link to download a copy of mine. Simply set your own USER and PASS:

<http://Valery.Letroye.be/BeatificaBytes/backupMySql.sh>

V.



parajesus

on **January 1, 2015 at 13:33** said:

Hey Folks,

For my DSM

```
/*
```

```
Model Name: DS214se
```

```
Current DSM version: DSM 5.1-5021 Update 2
```

```
DSM date: 2014/12/24
```

```
*/
```

I had to replace the the shell line in the script to:

```
#!/bin/ash
```

Apart from that minor change works magnificent!

Thanks Pal!



Valery Letroye

on **January 13, 2015 at 10:16** said:

Thx for sharing that info! I will fix my script.

I don't even know why I did set bash (except out of habit) :/
'ash' is indeed the default root shell...

The shebang line is only required if the script is used "as an executable".
In my case, I run it with the command sh /volume1/backup/backupMySql.sh (also out of habit)
So, the shebang line is ignored (as far as I know).



isaacku

on **April 2, 2015 at 04:53** said:

How can I restore databases over 1.8gb



Valery Letroye

on **April 18, 2015 at 14:14** said:

1.8GB is a limit of phpMyAdmin.

You can try, via a shell, to execute: `mysql -p -u[user] [database] < mysqldump`



Seren

on **September 9, 2015 at 02:07** said:

I seem to be having an issue with the DATESTAMP in that it adds a space at the end of the stamp, and prior to any other characters, corrupting the file. Any ideas on what could/would cause that? (scheduled task is running fine/without issues - other than the file unread, if I remove the datestamp information entirely, all is well.



Valery Letroye

on **October 13, 2015 at 20:26** said:

Sorry, no... this doesn't ring any bell with me :/



Matthias

on **January 14, 2016 at 20:22** said:

Hi great script! Thanks

Rewrote last part a bit to save DB's individually

```
databases=`mysql --user=$DB_USER --password=$DB_PASS -e "SHOW DATABASES;" | tr -d " | " | grep -v Database`  
for db in $databases; do  
if [[ "$db" != "information_schema" ]] && [[ "$db" != *_* ]]; then  
mysqldump --force --opt --user=$DB_USER --password=$DB_PASS --databases $db >  
$DIR/$DATESTAMP/dbbackup-$DATESTAMP-$db.sql  
gzip $DIR/dbbackup-$DATESTAMP-$db.sql  
fi  
done
```



parti78

on **March 25, 2016 at 07:49** said:

This script works for you DSM under 6?



Valery Letroye

on **March 28, 2016 at 18:40** said:

I just upgraded my NAS to DSM 6.0 and this script does not run anymore (out of the box). ~~I will investigate.~~
The reason is that "/usr/syno/mysql/bin/mysqldump" has been moved to "/bin/mysqldump". I did update my script and it works now fine.



parti78

on **March 28, 2016 at 19:08** said:

Thanks it works!



Dax

on **January 29, 2017 at 13:27** said:

Great stuff!! Thank you soooo much!



Valery Letroye

on **January 8, 2018 at 22:43** said:

For your information, new path of mysqldump when installed with:

MariaDB 5: /volume1/@appstore/MariaDB/usr/bin/mysqldump

MariaDB 10: /volume1/@appstore/MariaDB10/usr/local/mariadb10/bin/mysqldump



Clifford Too

on **January 8, 2019 at 13:38** said:

thank valery. your script work. thanks for the great job