

Can I use OUTFILE in Managed MySQL?

Yes it's possible on Elestio. Since we are running MySQL inside a docker container you will have to modify the stack to mount a new volume in the docker-compose. You can edit it with UPDATE CONFIG button from our dashboard

You will have to add a line like this in the mysql container

- ./files:/var/lib/mysql-files

```
1 version: '3'
2 services:
3   mysql:
4     image: mysql:${SOFTWARE_VERSION_TAG}
5     restart: always
6     command: mysqld --default-authentication-plugin=mysql_native_password --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci --gtid-mode=ON --enforce-gtid-consistency=ON
7     environment:
8       MYSQL_ROOT_PASSWORD: ${SOFTWARE_PASSWORD}
9     ports:
10      - 172.17.0.1:3306:3306
11     volumes:
12      - ./data:/var/lib/mysql
13      - ./files:/var/lib/mysql-files
14
15   pma:
16     image: phpmyadmin/phpmyadmin
17     restart: always
18     links:
19      - mysql:mysql
20     ports:
21      - "172.17.0.1:24581:80"
22     environment:
23       PMA_HOST: mysql
24       PMA_PORT: 3306
25       PMA_USER: root
26       PMA_PASSWORD: ${SOFTWARE_PASSWORD}
27       UPLOAD_LIMIT: 500M
28       MYSQL_USERNAME: root
29       MYSQL_ROOT_PASSWORD: ${SOFTWARE_PASSWORD}
30     depends_on:
31      - mysql
32
```

Then click on the button "Apply & restart"

After that you can use queries like this:

```
SELECT * FROM `help_topic` INTO OUTFILE '/var/lib/mysql-files/test1.csv';
```

and your file test1.csv will be present on the host vm in /opt/app/files/test1.csv

You can access it over SSH / SFTP / File Explorer / VS Code ...

If you want to use InFile to import CSV to a Table, you must add another option at end of row 6 (command) of the docker-compose file:

--local-infile=1

Then also modify the mysql-docker-cli.sh to add "--local-infile" at the end of it

Finally you can start the cli with:

./opt/app/mysql-docker-cli.sh;

from there you can now load data from CSV with a query like this:

use YOUR_DB_NAME_HERE;

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/**yourFile.csv**' INTO TABLE **yourTableNameHere**
FIELDS TERMINATED by ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

You should then see something like:

Query OK, 10634 rows affected, 65535 warnings (0.48 sec)

Records: 10634 Deleted: 0 Skipped: 0 Warnings: 251949

Revision #4

Created 13 April 2023 18:43:36 by Joseph Benguira

Updated 12 June 2023 14:16:52 by Joseph Benguira