Liftians Database Deployment Procedure

Liftians Inc. v0.1

Scope

This document details the process of creating a MySQL database backup and deploying it to a production server. This assumes you are familiar with Linux command-line environment and database users and privileges are set appropriately. You should also be comfortable with a secure shell (SSH) client for connecting to a server. You should be familiar with MySQL Workbench as well. There should be no errors involved in backing up the database and deploying the database. The database username used to make the backup on the development server and the username used to restore the backup on the production server should already be created with correct permissions. In this guide the *System* username and *Database* usernames are the same (liftians). This guide also assumes the production server is already configured and in place in the physical production environment (client server room). The server should be preconfigured with Ubuntu *System* users, MySQL *Database* users, VPN access, and DNS forwarding for the server's name. The example used in this document will be copying the database *Liftman_CustomerA* from the Liftians development server 'SD1.liftians.com' to the practice production server 'CustomerADev.liftians.com'.

TL;DR

- Command Line Interface (CLI) experience
- Familiarity with SSH Clients (ex. PuTTY)
- Familiarity with MySQL Workbench
- Creation of MySQL Database Users and Tables (liftians)
- Creation of Ubuntu System Users (liftians)
- VPN client access
- Dev Server: SD1.liftians.com
- Prod Server: CustomerADev.liftians.com

Document Revisions

Version	Author	Date	Changes
v0.1	Nick Baker	3/11/2022	First draft
V1.0	Jonathan Huang	4/25/2023	Version 1.0

Table of Contents

1. Connect to Dev Server

Establish a Secure Shell (SSH) to the Dev Server (SD1.liftians.com) using the appropriate credentials:

- User: liftians
- PW: liftians**** (last 4 characters are not shown here)

```
login as: liftians
liftians@SD1.liftians.com's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-163-generic x86_64)
```

2. Navigate to Backup Location

The backup can be set to output to the backup directory by specifying /home/Liftians/mysql/backup for the output or by navigating there first. We will navigate to the directory first to simplify the backup. After connecting change directory by typing: **cd mysql/backup/**

```
liftians@SD1.liftians.com:~$ cd mysql/backup/
liftians@SD1.liftians.com:~/mysql/backup$
```

3. Create Database Backup

Use mysqldump command-line utility to create the backup. Since we already navigated to the backup location, we don't need to specify a directory for the output as it will dump it to the current working directory. We already know the database name we want to back up: 'Liftman_CustomerA'. Create the backup by typing the following:

mysqldump --routines -u liftians -p --no-tablespaces -y Liftman_CustomerA > Liftman_*date*.sql

```
liftians@SD1.liftians.com:~/mysql/backup$ mysqldump --routines -u liftians -p --
no-tablespaces -y Liftman_TheFeed > Liftman_20220311-1.sql
Enter password:
liftians@SD1.liftians.com:~/mysql/backup$
```

The username and password used here is for the *database user* since we are interacting with the database. The options passed are as follows:

- --routines – this will back up the stored procedures and functions
- -u – the database username
- -p – requires database user password
- --no-tablespaces -y – we do not need to dump the tablespace information

We see that there were no errors making the backup. We can see the outputted file by typing **ls**:

```
liftians@SD1.liftians.com:~/mysql/backup$ ls
FULLBK20180618.sql
FULLBK.sql
Liftman_20220311-1.sql
Liftman_Hansom.sql
Liftman_Phicomm_2018_06_06_17_hourly_bk.sql
```

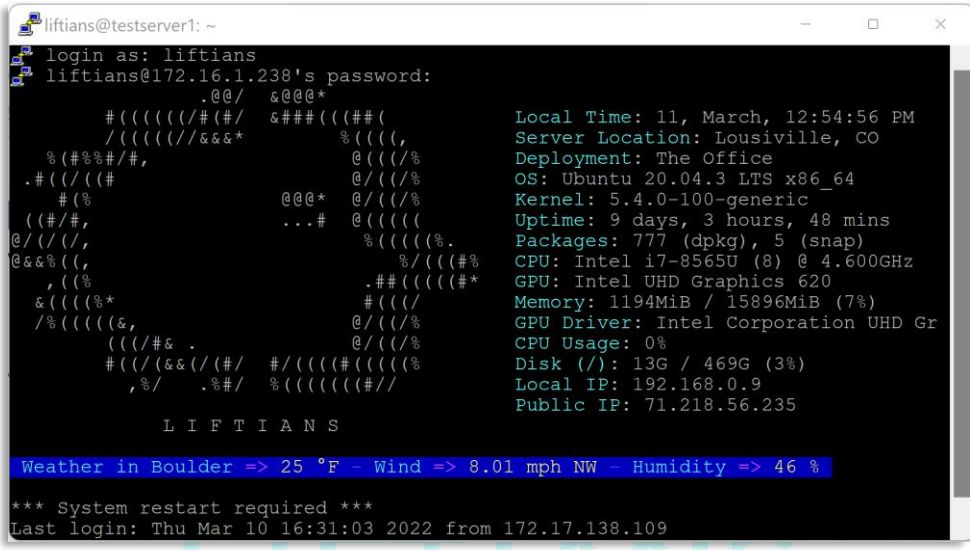
4. Copy the File to Production Server

We will use scp (secure copy protocol) to send the file from the Dev Server to the Production Server. You can do this from either machine. We will do it from the perspective of the Production Server.

4.1. Log in to Production Server

Establish a Secure Shell (SSH) to the Production Server (CustomerADev.liftians.com, currently named *testserver1*) using the appropriate credentials:

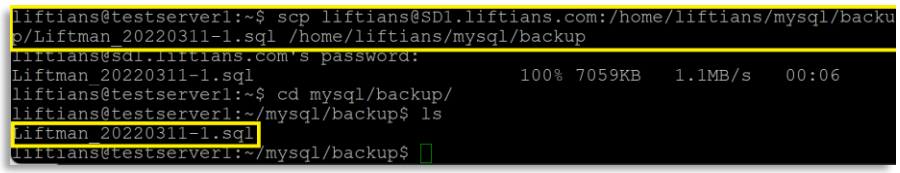
- User: liftians
- PW: liftians**** (last 4 characters are not shown here)



4.2. scp Backup from Dev Server to Production Server

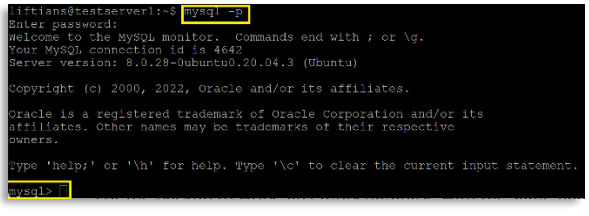
Copy the backed-up database file with the following:

scp liftians@SD1.liftians.com:/home/liftians/mysql/backup/Liftman_20220311-1.sql /home/liftians/mysql/backup



4.3. Create Database Name on Production Server

For the backup to work, an empty database with the same name needs to exist first. Log in as the *database* user to MySQL by typing: **mysql -p**



Now that we are logged in to MySQL we will create the new empty database. Type the following to create the new empty database: **CREATE DATABASE Liftman_CustomerA;**

```
mysql> CREATE DATABASE Liftman_TheFeed;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Liftman_Assembly   |
| Liftman_TheFeed    |
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```

With the empty data base created, we can restore the backup to it. Exit the mysql prompt by pushing Ctrl+D. Navigate to the /home/liftians/mysql/backup folder.

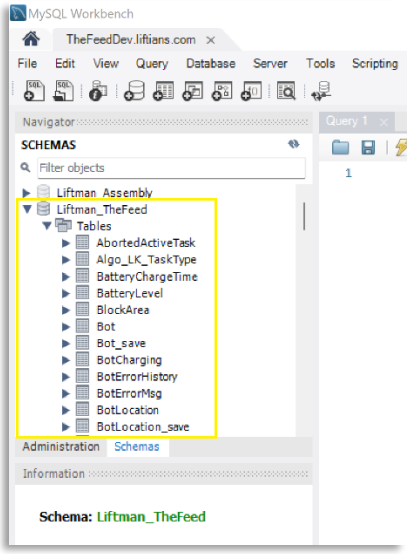
```
liftians@testserver1:~$ cd mysql/backup/
```

4.4. Restore Database on Production Server

Restore the backup file 'Liftman_20220311-1.sql' using the *database* user and password by typing: **mysql -u liftians -p Liftman_CustomerA < Liftman_20220311-1.sql**

```
liftians@testserver1:~/mysql/backup$ ls
Liftman_20220311-1.sql
liftians@testserver1:~/mysql/backup$ mysql -u liftians -p Liftman_TheFeed < Lift
man_20220311-1.sql
Enter password:
liftians@testserver1:~/mysql/backup$
```

We see there are no errors restoring the backup. The database structure can be verified in MySQL Workbench:



The process is complete, the database backup has been restored to the Production server.