Cloud SQL MySQL Upgrade Tool User Guide

Deployment Guide

Date: January , 2024

Authors: Rani Singh Siddhant Verma Piyush Sarraf

Reviewer: Swati Aggarwal

Prepared for: Google

Document Type: Deployment Guide

Document Version

Name	Date	Reason for changes	Version	Approved by	Approval Date	Created by	Created Date
First Draft	17/01/2024	Initial Draft	1.0	Google Team		Google Team	17/01/2024
Second Draft	31/01/2024	Updated Installation Steps					
	22/05/2024	Added steps for binary file execution					

Contents

1. Overview of Deployment Guide and Tool

1.1 Objective of Deployment Guide

This document is created to help users run the Cloud SQL Upgrade Tool for MySQL MVU with E+ . It is a step-by-step guide to run the tool.

1.2 About the MySQL Upgrade Tool

This tool is developed to enable automated assessment and upgrades of MySQL databases for MYSQL 5.7.x to 8.0.31 or above with Enterprise edition to Enterprise Plus.
The approach here is to do In-place major version upgrade with automated assessment that can fix known issues and guide during the overall upgrade process.
It is basically a 2 step process first being Major Version Upgrade followed by Enterprise + upgrade.

This tool can be run by a user that has privileges to run mysql checks and to be able to upgrade MySQL databases.
It generates a PDF report with information related to the extensive sanity checks performed to evaluate the compatibility, data integrity for upgrading MySQL 5.7 instance to 8.0.31 and to capture a baseline performance.

The is a combination of Python Scripts and gcloud commands that runs sanity checks on MySQL database and upgrades MYSQL 5.7 instance to 8.0.31 version.

Following are the sanity checks that are being performed:

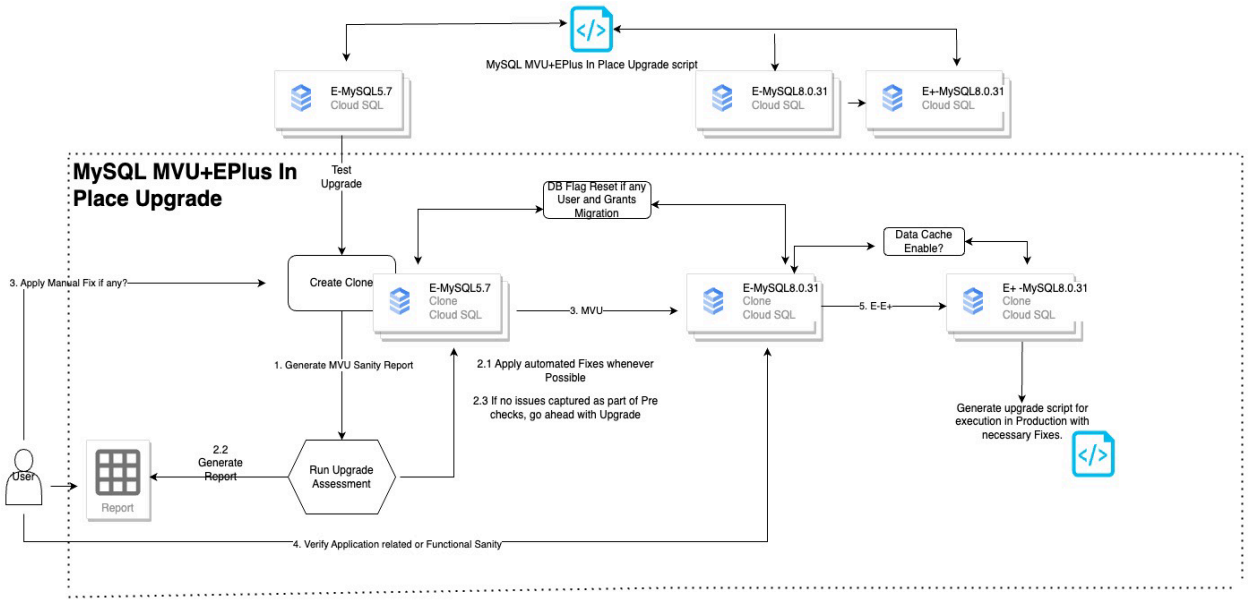
- **Check for server upgrade** - The util.checkForServerUpgrade() function is an upgrade checker utility that enables you to verify whether MySQL server instances are ready for upgrade. The upgrade checker utility carries out the automated checks that are relevant for the specified target release,and advises you of further relevant checks that you should make manually.

- **Column Name size check** - Prior to MySQL 8.0, users could create views with explicit column names up to 255 chars. To adhere to the maximum length of column name, views having explicit column name greater than 64 chars is not supported in MySQL 8.0
- **Constraint Name size check** -In MySQL 8.0, tables with foreign key constraints where the constraint name exceeds 64 chars are not supported in order to adhere to the maximum identifier length of database objects.
- **Transactional Data Dictionary check** - The transactional data dictionary(DD) support is introduced in MySQL 8.0 for which several new DD tables are created in the mysql schema. Hence user tables with the conflicting names in the mysql schema should be dropped or explicitly renamed prior to upgrade.
- **MySQL Check upgrade table** - It Performs table level checks to identify issues that can cause failure or features deprecated in MySQL 8.0.
- **Database Flags check** - Checks Database Flags set for the particular cloud sql instance
- **Disk space check** - Before running a major version upgrade, ensure that you have more than 100K of memory for each table. We will need to measure the total number of tables and increase memory as per it. It should be dynamically fixed.

2. Tool Functionality

The Cloud SQL upgrade tool is built to perform sanity checks on the MySQL databases and generate reports with detected problems along with automated scripts to fix some of the issues. Once it passes all the sanity checks then Major Version Upgrade from MySQL 5.7 to 8.0.31 will be carried out followed by Enterprise plus upgrade.

Architecture



1. This tool will need below packages to be installed.
    a. Python Packages
        i. wkhtmltopdf
        ii. mysql.connector
        iii. Pdfkit
        iv. Jinja2
        v. google-cloud-secret-manager
        vi. matplotlib
2. Input to the Framework are,
    a. MySQL DB Host
    b. Port
    c. Instance Name

d. User id and password (Should have privileges to run mysql sanity checks)

3. Output Generate by the tool are,
    a. #1 PDF file - assessment report
    b. 3 scripts to fix the identified issues for DB flags, Data dictionary tables, Grants scripts.

3. Prerequisites

3.1 Important highlights of the Tool

★ The tool runs on the following OS versions **Debian 11, MacOS.**
★ Reach out to your Google account teams for access to the repo
★ To ensure that the tool can run sanity checks for you, verify that the server where it is being deployed has access to your MySQL instance.

3.2 Complete information to run the tool

1. The tool can run on any supported VM that conforms to prerequisites listed in **Section 3.1**, and fulfills following prerequisites:
    1.1. Connectivity to MySQL instance
    1.2. DB user with privileges to perform sanity checks and upgrade.
2. The tool initially requires **< 200mb** of space
3. This tool has been tested on the MySQL 5.7 version.
4. This tool runs on the following **Operating systems**:
    4.1. Debian 11 and above
    4.2. MacOS
5. After a successful tool execution, a PDF report will be generated at the location: `./storage/reports/mysql-report.pdf`.

6. After a successful tool execution, 3 scripts will be generated to automatically fix identified issues at the location:
   ./storage/scripts/afterupgrade_databaseFlagScript.sql
   ./storage/scripts/beforeupgrade_datadictionarytables.sql
   ./storage/scripts/afterupgrade_grants.sql

3.3 User Input Requirements

1. The tool needs the below permissions to run the code and generate the PDF report.
   1.1. MySQL admin user to run sanity checks on the database and to execute MVU and E+ Upgrade.
   1.2. Or the user in the MySQL database should have below mentioned privileges to be able to proceed with an upgrade

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP,
RELOAD, SHUTDOWN, PROCESS, REFERENCES, INDEX, ALTER, SHOW
DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE,
REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW
VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT,
TRIGGER, CREATE TABLESPACE on *.* to 'user'@'localhost'
WITH GRANT OPTION;
```

4. Installation Steps

Debian -

1. **Request access -**
   - Request the account team to submit an access request for the binary file.

2. **Install Wkhtmltopdf package**: Install the wkhtmltopdf package using the following command -

```
Unset
For linux -
sudo apt-get install wkhtmltopdf

For Mac -
brew install wkhtmltopdf
```

3. **Download MySQL Shell** - Execute the below command to install MySQL Shell for linux.

```
Unset
wget
https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell_8.0.31-1debian11_am
d64.deb
wget
http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2_
amd64.deb
sudo dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb
sudo dpkg -i mysql-shell_8.0.31-1debian11_amd64.deb

Note: For other OS refer to this page.
```

4. **Download MySQL Client** - Install MySQL Client using the following command -

```
Unset
For linux -
sudo apt-get update
sudo apt-get install default-mysql-client

If encountered issue "E: Unmet dependencies. Try 'apt --fix-broken install'
with no packages (or specify a solution)."

Then run below commands to resolve.
sudo apt --fix-broken install
sudo apt-get install default-mysql-client

For Mac -
brew install mysql-client
brew install mysql
```

5. **Download files**: Download the binary and config file, using the gsutil command below.

```
Unset
gsutil -m cp -r gs://cloudsqlupgrade/binaries/mysql . && cd mysql
```

6. **Setup gcloud auth login**: To execute gcloud commands, setup the auth login and project_id -

```
Unset
gcloud auth login
gcloud config set project <PROJECT_ID>
gcloud auth application-default login
```

7. **Download gcloud alpha component** - Install gcloud alpha component using the following command -

```
Unset
gcloud components install alpha
```

8. **Follow the steps to install proxy server to be able to connect Cloud SQL MySQL instance**:

   https://cloud.google.com/sql/docs/mysql/connect-instance-auth-proxy#install-proxy

9. **Populate the config file:** Add the respective values to the **config.json** file.

```
Unset
{
```

```
    "projectId": "",
    "instanceId": "",
    "user": "",
    "secretId": "",
    "password": "",
    "machineType": "",
    "runAssessment": "",
    "utilityChecker": "",
    "cloneSqlInstance": "",
    "replicaUpgrade": "",
    "enterpriseUpgrade": "",
    "enterprisePlusUpgrade": "",
    "transactionalDataDictionaryScript": "",
    "databaseFlagScript": "",
    "grantPermissionsScript": ""
}
```

Note: Some of parameter definition are as follows:
projectId: Project ID where Cloud SQL instance is hosted.
instanceId: Instance ID of Cloud SQL instance.
user: database user with required permissions
secretId: ID of file in Secret Manager where database user password is stored. Either provide the user password in secret manager(enter the secretID) or as a plain text in Password variable. Tool will pick either of these.
password: database user password(in plain text). If you have provided secretId above then leave this blank.
machineType: Machine type of Cloud SQL Enterprise Plus Edition. Select required machine type from below image.
Ex: db-perf-optimized-N-2
runAssessment: [Yes/No], if choosen "Yes", will run assessment on the instance and also generate Grant Script.
utilityChecker: [Yes/No], if choosen "Yes", will run Mysql Utility Checker on the instance, else would run checks mentioned in Mysql Utility Checker independently on the instance. This is recommended in case where instance has very large no. of databases and tables(>512,000). Refer to the limitations here.
cloneSqlInstance: [Yes/No], if choosen "Yes", will make a clone of current instance and perform upgrade on that instance, else perform upgrade on current instance.
replicaUpgrade: [Yes/No], if choosen "Yes", will upgrade all the replicas of the primary instance along with it, else will only upgrade the primary instance.

```
enterpriseUpgrade: [Yes/No], if choosen "Yes", will perform a Major Version Upgrade.
enterprisePlusUpgrade: [Yes/No], if choosen "Yes", will perform an upgrade to Enterprise Plus.
transactionalDataDictionaryScript: [Yes/No], if choosen "Yes", the same named user table and system table, will be renamed as tablename_upgrade.
databaseFlagScript: [Yes/No], if choosen "Yes", script will check and edit the flags, if modifiable.
grantPermissionsScript: [Yes/No], if choosen "Yes", will grant the same permission to the Cloud SQL user, which was before Major Version Upgrade.
```

Choose any machine type from below list :

Machine types for Cloud SQL Enterprise Plus edition instances For Cloud SQL Enterprise Plus edition instances, machine types are predefined as follows:

Enterprise plus machine type	vCPUs	Memory (GB)	Local SSD
db-perf-optimized-N-2	2	16	375
db-perf-optimized-N-4	4	32	375
db-perf-optimized-N-8	8	64	375
db-perf-optimized-N-16	16	128	750
db-perf-optimized-N-32	32	256	1500
db-perf-optimized-N-48	48	384	3000
db-perf-optimized-N-64	64	512	6000
db-perf-optimized-N-80	80	640	6000
db-perf-optimized-N-96	96	768	6000
db-perf-optimized-N-128	128	864	9000

10. To Execute the upgrade script follow steps below -

10.1. Add the execution permission to binary file:

```
Unset

    chmod +x cloudsql_upgrade
```

10.2. Run the binary file:

```
Unset
    ./cloudsql_upgrade
```

10.3. Verify logs generated at :

```
Unset
./storage/logs/app.log
```

10.4. Run the following command to start the Cloud SQL Proxy Server

```
Unset

"Open a new session in the terminal and run the following command - "
./cloud-sql-proxy {connectionName}

Note : Open a new terminal and execute the command given above and let the
session run.
Example command : "./cloud-sql-proxy adt-rani-testing:us-central1:mysql-5-7"
```

10.5. Go back to the first terminal and provide your Cloud SQL Auth Proxy Host:

```
Unset
    Please provide your Cloud SQL Auth Proxy Host: 127.0.0.1

Note: By default the host is 127.0.0.1 and port is 3306. Incase if it is
different provide appropriately.
```

10.6. Provide your Cloud SQL Auth Proxy Port:

```
Unset
    Please provide your Cloud SQL Auth Proxy Port: 3306

Note: By default the host is 127.0.0.1 and port is 3306. Incase if it is
different provide appropriately.
```

10.7. Please make sure all the problems are resolved, which are detected during sanity checks:

```
Unset
Prompt: "Please check and resolve if any problems detected during assessment"
Confirm if all the issues are resolved and good to proceed with clone and
upgrade?(yes/no) -
```

Now the script will create a clone or not based on the input provided in the config file.
Note : Creating a clone of the instance might take a few minutes.

10.8. As we will now use the cloned instance for further steps , go ahead and terminate the 2nd terminal which we used at step 11.2 for proxy connection to the main instance.

10.9. Now create a new proxy session for the cloned instance.

```
Unset

    Clone Done for mysql-5-7
```

```
   ************************************************************
   Open a new session in the terminal and run the following command -
   ./cloud-sql-proxy adt-rani-testing:us-central1:mysql-5-7-clone

Note : Open a new terminal and execute the command given above and let the
session run. Pick the cloned instance name from the output above.
Example command : "./cloud-sql-proxy
cloudsql-testing:us-central1:mysql-5-7-clone"
```

10.10. Go back to terminal and provide your Cloud SQL Auth Proxy Host and port:

```
Unset
   Please provide your Cloud SQL Auth Proxy Host: 127.0.0.1

Note: By default the host is 127.0.0.1 and port is 3306. Incase if it is
different provide appropriately.

   Please provide your Cloud SQL Auth Proxy Port: 3306

Note: By default the host is 127.0.0.1 and port is 3306. Incase if it is
different provide appropriately.
```

10.11. Please verify if the clone and upgrade ran successfully or not.

10.12. After the successful upgrade please make sure the instance is ready for Enterprise Plus upgrade

```
Unset
Confirm if the instance is ready for Enterprise Plus Upgrade?(yes/no) -
```