Migrating Data from MariaDB to Azure for MariaDB



Prepared by

**Bram Pahlawanto**

Solution Architect, Data SQL Ninja Engineering Team

[bram.pahlawanto@microsoft.com](mailto:bram.pahlawanto@microsoft.com)

Nov-2019

Contents

[Introduction 3](#_Toc859808)

[Conventions and Assumptions 3](#_Toc859809)

[Pre-requisites 4](#_Toc859810)

[Installation 5](#_Toc859811)

# Introduction

This article explains how to use the python script expimpmysql.py to migrate data from any regular MariaDB Database to Azure for MariaDB

The features of this scripts are:

1. Fully configurable
2. Extract data to a “tab-delimited” format within a gzip file
3. Import the extracted data that are in gzip file to a target database
4. Configurable parallelism, no of chunks, character set, maximum rows per csv file
5. Selectable tables or all tables
6. Obfuscated password is stored in a config, so it doesn’t require to re-enter password when performing data migration
7. Generated a logfile

# Conventions and Assumptions

The following variables are used for ease of installation

* Server Name: Linux1
* Config file: mysqlconfig.ini
* Linux prompt: #
* Sample mysqlconfig.ini

[export]  
servername = archdocker

port = 3306

username = bram

database = bram

charset = latin1

rowchunk = 100000

maxrowsperfile = 2000000

tables = all

parallel = 5

password = Oa{u

[import]

servername = archdocker

port = 3306

username = bram2

database = bram2

charset = latin1

parallel = 5

rowchunk = 100000

tables = all

password = Oa{uC

# Pre-requisites

Any Linux distros (distributions) can be used to run this script as long as python (3.6 or later) is installed

Before performing the installation, the following pre-requisite steps must be satisfied:

* Python 3.6 or later
* Python-pip3 is installed
* Pymysql library for python is installed
* Connection to the source database and target database can be performed and the designated ports on both source and target are opened
* If necessary, run this script on a dedicated machine which has few number of CPU and few giga bytes of memory

# Installation

After the pre-requisites have been satisfied, execute the following steps:

1. Copy the script file expimpmysql.py in a directory and ensure the directory has writable permission to the user who will be running the script
2. Copy the config file mysqlconfig.ini into the same location as the above script
3. Modify the above config file with the correct information such as servername, username, database, charset, etc (see the sample config above)
4. Speceficially for password, must be left empty E.g: password=  
   NOTE: this will automatically prompt user for password only for the first time, once the password is entered, it will be obfuscated and stored within the above config file

# Run the script

This script is actually self-explanotry when it is run without parameter, it will automatically display the help information

[root@centospgsql mariadb]# ./expimpmysql.py

Usage:

expimpmysql.py [OPTIONS]

General options:

-e, --export export mode

-i, --import import mode

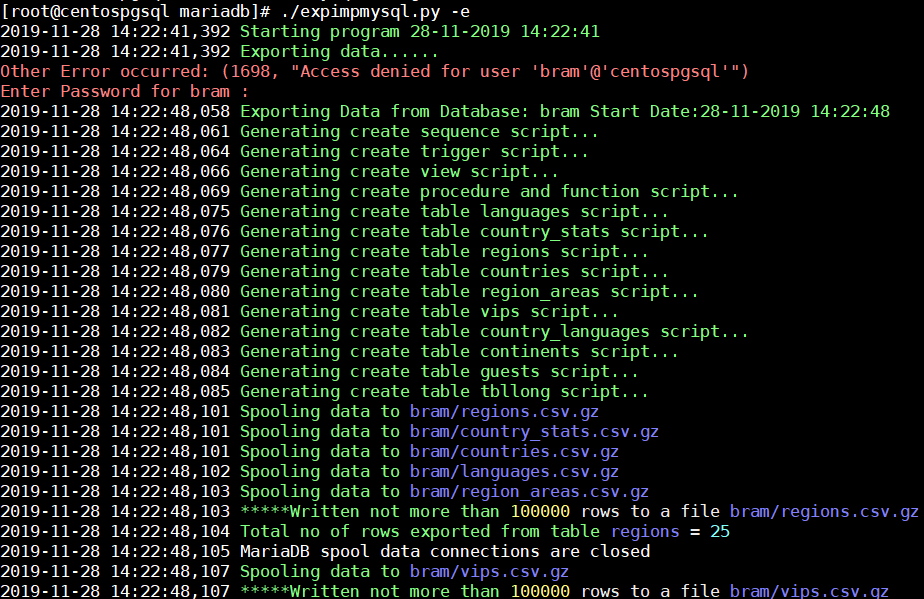
-s, --script generate scripts

-d, --dbinfo gather DB information

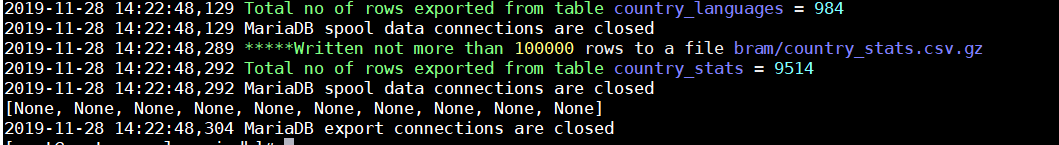
-a, --all-info gather All information from information\_schema

-l, --log= INFO|DEBUG|WARNING|ERROR|CRITICAL

The following is the extract/export process, the extracted data will be stored in a directory that is named the same as the source database

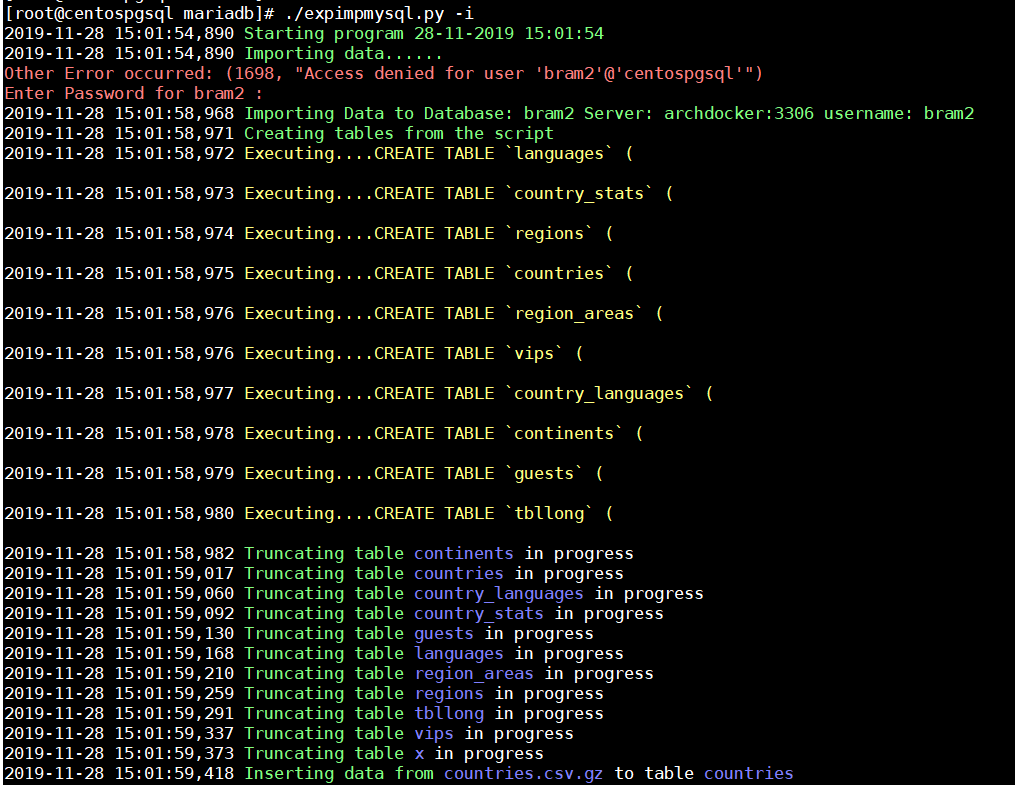
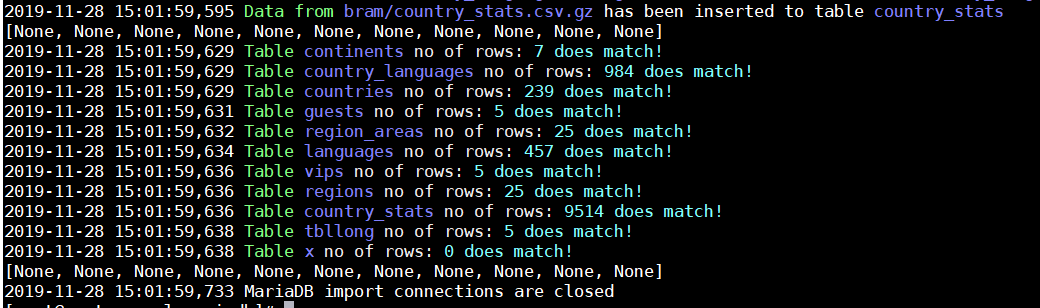
[root@centospgsql mariadb]# ./expimpmysql.py -e  


….

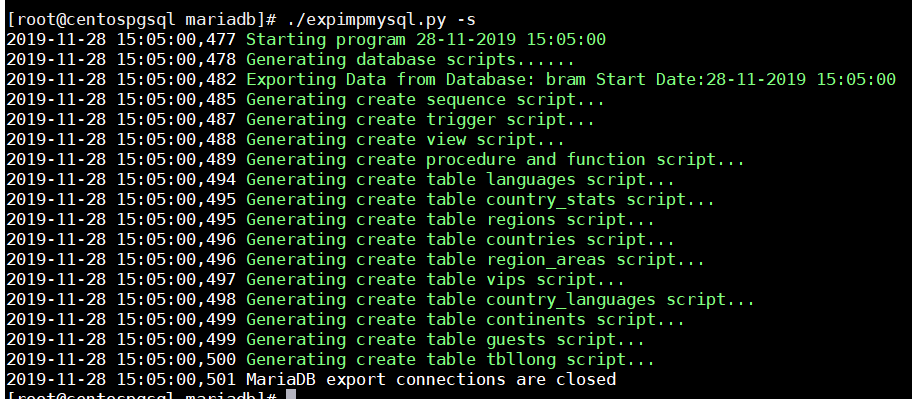


Once the data has been extracted, then perform import process as follow

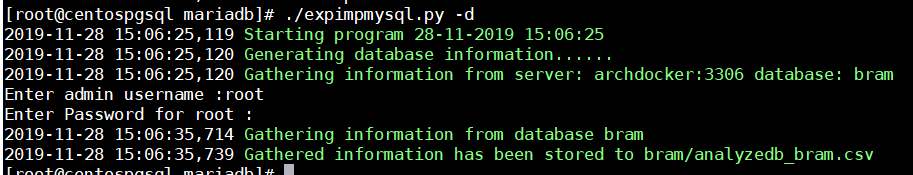
[root@centospgsql mariadb]# ./expimpmysql.py -i

  
……  


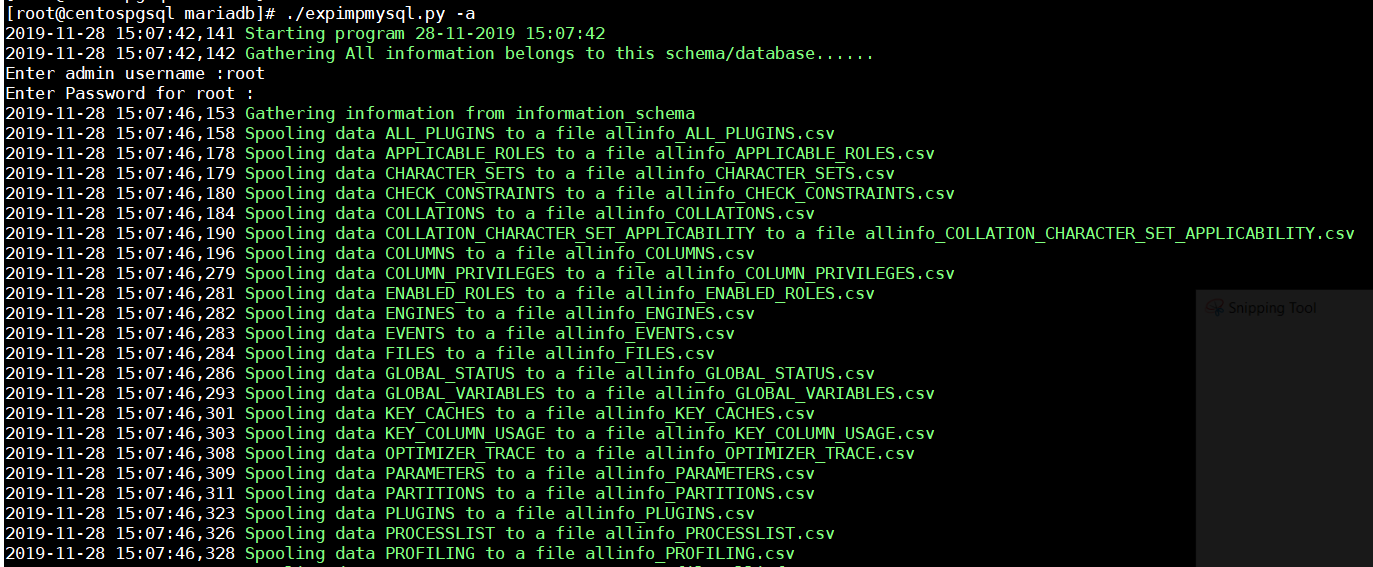
To generate scripts then do the following:  
[root@centospgsql mariadb]# ./expimpmysql.py -s



Gathering database info: (it requires super-user priv)



Gathering all information from information\_schema:



….

….

