# Migrating data from On-Premises MySQL to Azure Database for MySQL using Azure DMS (.NET)

## Problem Statement:

In this documentation you will learn how to migrate On-premises MySQL Database to Azure database for MySQL using DMS. Here I have taken demo sample database to migrate with minimal downtime.

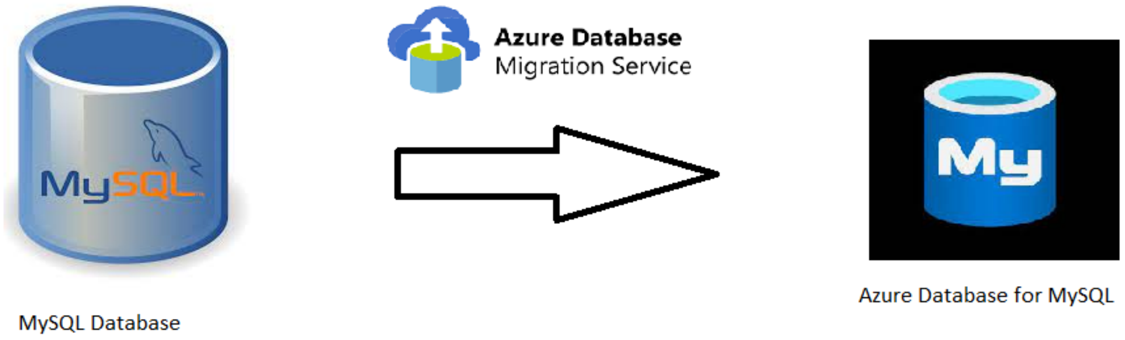
- Azure Database for MySQL:

This is the fully managed MySQL database engine based on the stable version of MySQL community edition. This relational database as a service (DBaaS), hosted on the Azure cloud platform, falls into the industry category of PaaS.

- Azure Database Migration Service (DMS):

Azure Database Migration Service is a tool that helps you simplify, guide, and automate your database migration to Azure. Easily migrate your data, schema, and objects from multiple sources to the cloud at scale.

## Solution Architecture:



## Solution:

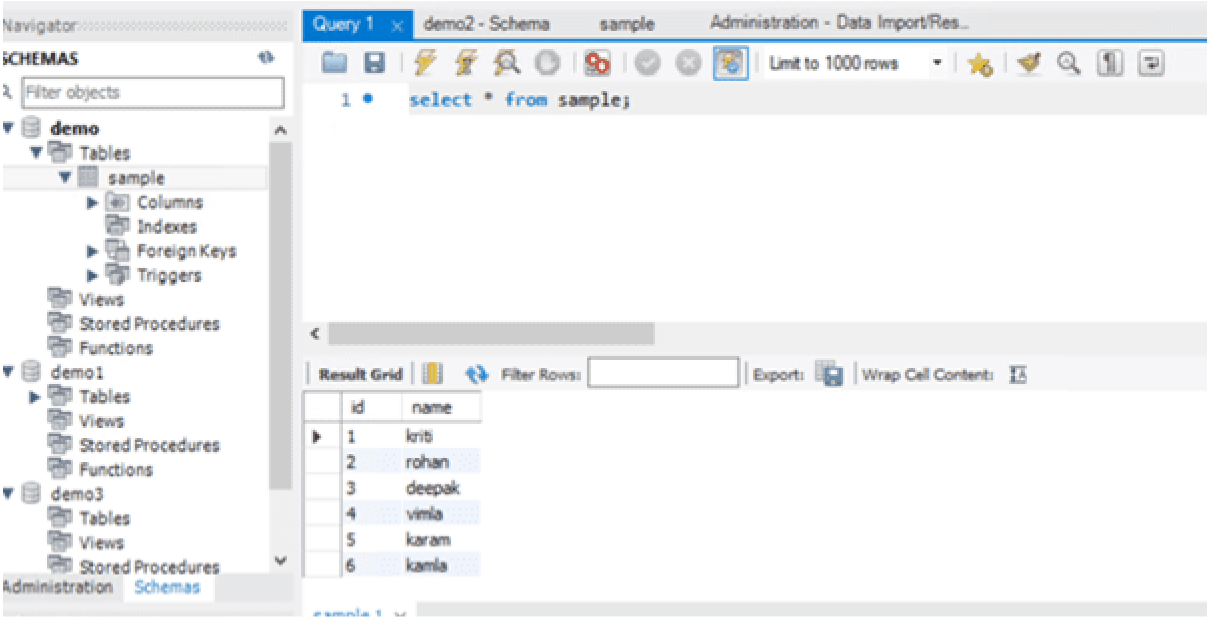
- Pre-requisites:

1. Create Azure database for MySQL.
2. Create a virtual machine.
3. Create DMS.
4. Install MySQL workbench.
5. Both Azure database for MySQL and on-premises MySQL should be of same version. (i.e., 5.7 or above).
6. Add client IP address and VM IP address to the Azure Database for MySQL.

7. Add 443 outbound port and 3306 inbound port on virtual machine.
8. Configure the DNS name of the VM.

- <u>Steps to follow:</u>

1. Download MySQL workbench on Virtual machine.
2. Create a database and a table on your local host and fill some entries in the table.



```
1 select * from sample;
```

| id | name |
|---|---|
| 1 | kriti |
| 2 | rohan |
| 3 | deepak |
| 4 | vimla |
| 5 | karam |
| 6 | kamla |

3. Create Azure database for MySQL using portal.
4. Connect azure database for MySQL with Workbench.
5. Go to the following directory on cmd.

```
C Drive-> program files -> MySQL -> MySQL server 8.0 -> bin
```

6. Write the following command on command prompt to give all the privileges to your user.

```
Mysql> CREATE USER ‘monty’@’localhost’ IDENTIFIED BY ‘some_pass’;
mysql> GRANT ALL PRIVILEGES ON *.* TO ‘monty’@’localhost’
->     WITH GRANT OPTION;
mysql> CREATE USER ‘monty’@’%’ IDENTIFIED BY ‘some_pass’;
mysql> GRANT ALL PRIVILEGES ON *.* TO ‘monty’@’%’
->     WITH GRANT OPTION;
```

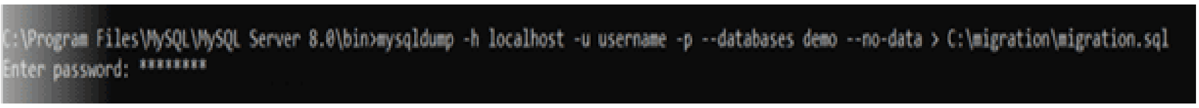
7. Use **mysqldump** command to do schema migration.
-h: **localhost** (You can get this IP from VM)
-u: **username**
-p: **Password**
–databases: **demo**

```
mysqldump -h [localhost] -u [username] -p[password] –databases [db name] –no-data >

[schema file path]
```

Example:

```
mysqldump -h localhost -u username -p –databases demo –no-data > C:\migration\sample.sql
```



```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -h localhost -u username -p --databases demo --no-data > C:\migration\migration.sql
Enter password: ********
```

8. To import schema to Azure Database for MySQL target, run the following command:

-h: servername.mysql.database.azure.com

-u: mysqladminuser@servername

-p: Password of your MySQL Server

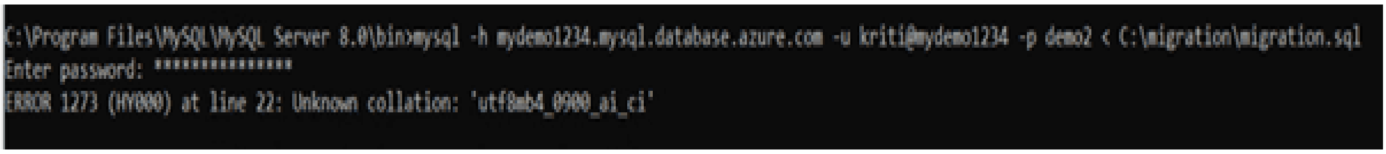
–databases: demo

```
mysql -h [servername] -u [username] -p[password] [database]< [schema file path]
```

Example:

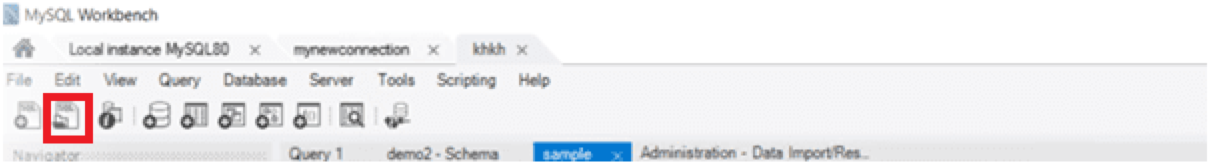
```
mysql -h mydemoservernm.mysql.database.azure.com -u myadmin@mydemoservernm -p demo <

C:\migration\sample.sql
```



```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -h mydemo1234.mysql.database.azure.com -u kriti@mydemo1234 -p demo2 < C:\migration\migration.sql
Enter password: ***************
ERROR 1273 (HY000) at line 22: Unknown collation: 'utf8mb4_0900_ai_ci'
```

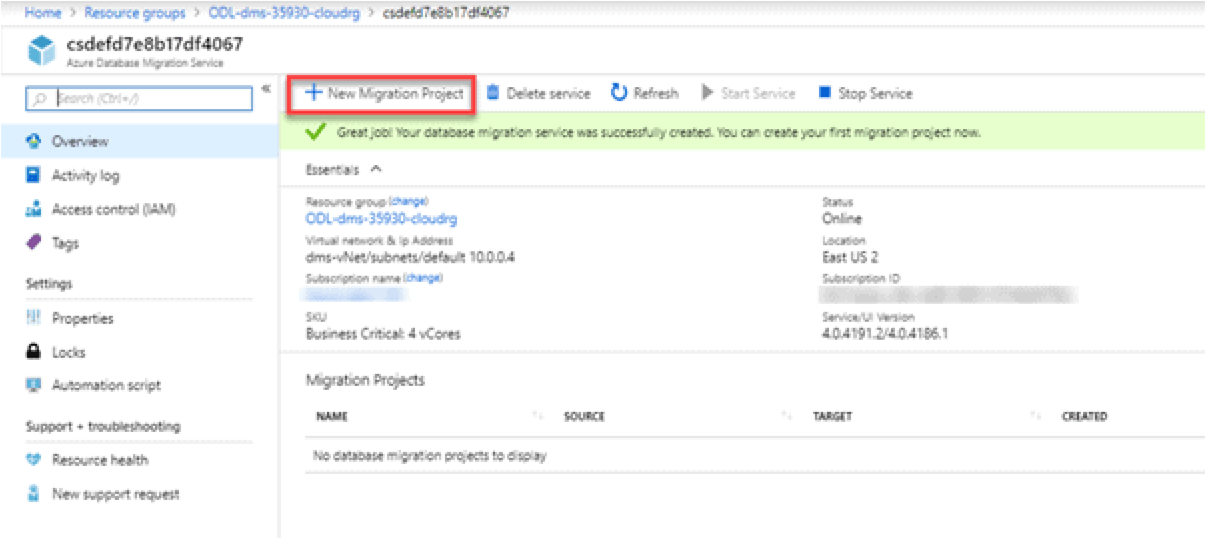
9. If you get the above collation error, then follow the below step:



- Click on the above red mark option
- Select the .sql file created before and open it.
- Now where ever ‘utf8mb4_0900_ai_ci’ is written, replace it with ‘utf8mb4_unicode_ci’ and save it.
- Now again run the above command

10. Now, the schema is migrated to Azure database for MySQL.

11. Create database migration service instance and then create a migration project on Azure Portal.



Great job! Your database migration service was successfully created. You can create your first migration project now.

Resource group (change)
ODL-dms-35930-cloudrg

Status
Online

Virtual network & Ip Address
dms-vNet/subnets/default 10.0.0.4

Location
East US 2

Subscription name (change)
[redacted]

Subscription ID
[redacted]

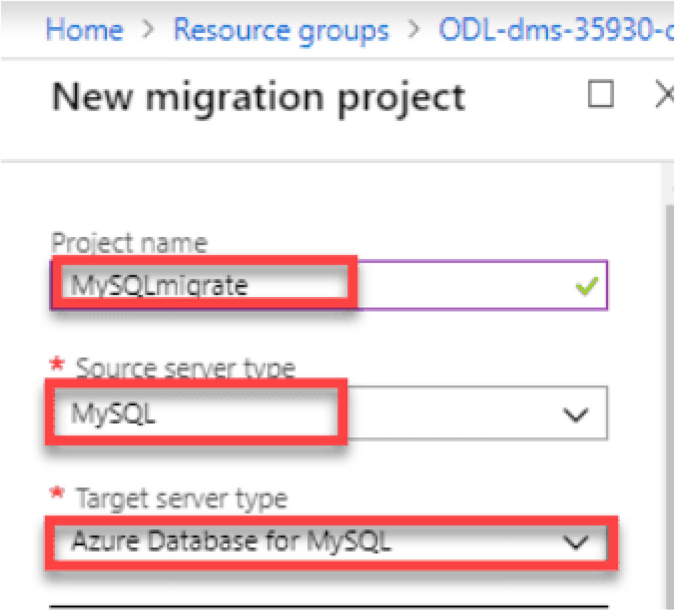
SKU
Business Critical: 4 vCores

Service/UI Version
4.0.4191.2/4.0.4186.1

Migration Projects

| NAME | SOURCE | TARGET | CREATED |
|---|---|---|---|
| No database migration projects to display | | | |

12. Select source as MySQL and destination as Azure database for MySQL.

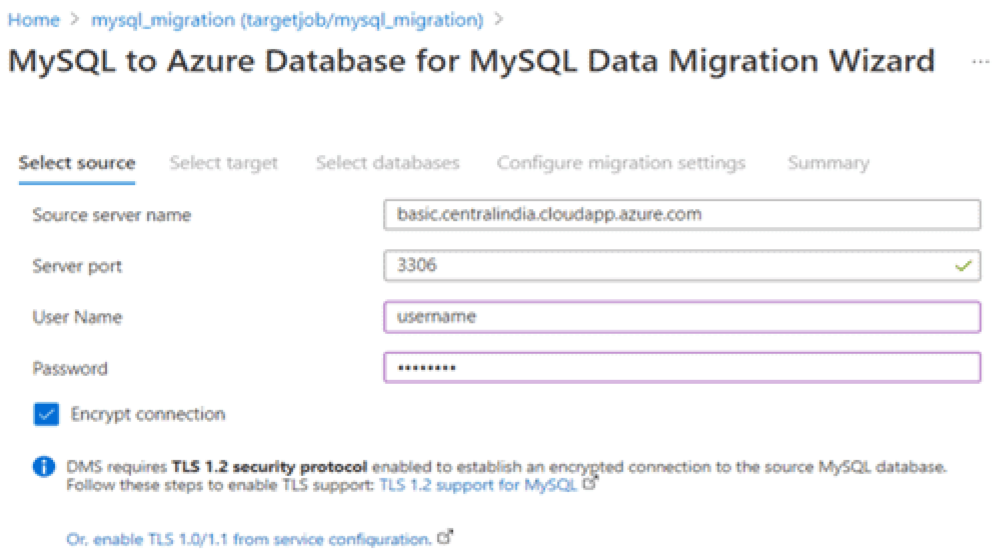


Project name
MySQLmigrate

* Source server type
MySQL

* Target server type
Azure Database for MySQL

13. On the Add Source Details screen, specify the connection details for the source MySQL instance. You have to add details of your on-premises MySQL server. Select Save
    - Source server name: DNS name of your VM
    - Server Port: 3306
    - Username: username
    - Password: Password


Home > mysql_migration (targetjob/mysql_migration) >
Select source
Select target
Select databases
Configure migration settings
Summary

MySQL to Azure Database for MySQL Data Migration Wizard

Source server name: basic.centralindia.cloudapp.azure.com

Server port: 3306

User Name: username

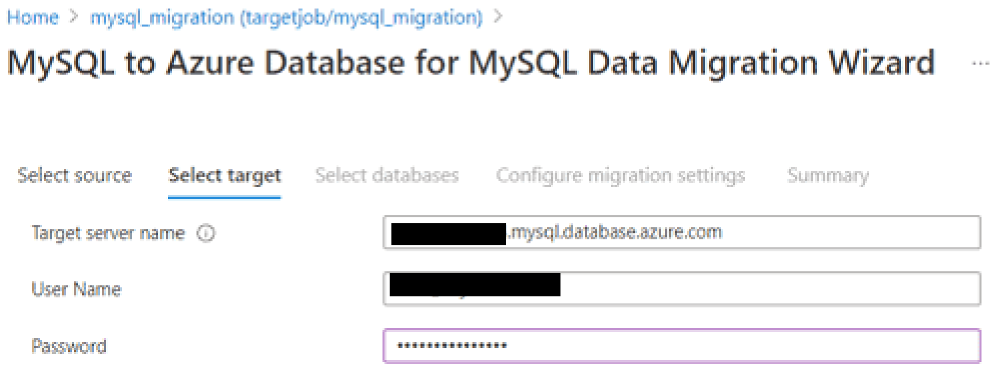
Password: •••••••••

☑ Encrypt connection

DMS requires **TLS 1.2 security protocol** enabled to establish an encrypted connection to the source MySQL database. Follow these steps to enable TLS support: TLS 1.2 support for MySQL

Or, enable TLS 1.0/1.1 from service configuration.

14. On the Target details screen, specify the connection details for the target Azure Database for MySQL server, which is the pre-provisioned instance of Azure Database for MySQL to which the Demo schema was deployed by using mysqldump.
    - Target Server name: servername.mysql.database.azure.com
    - Username: mysqladminuser@servername
    - Password: Password of your MySQL server


Home > mysql_migration (targetjob/mysql_migration) >
Select source
Select target
Select databases
Configure migration settings
Summary

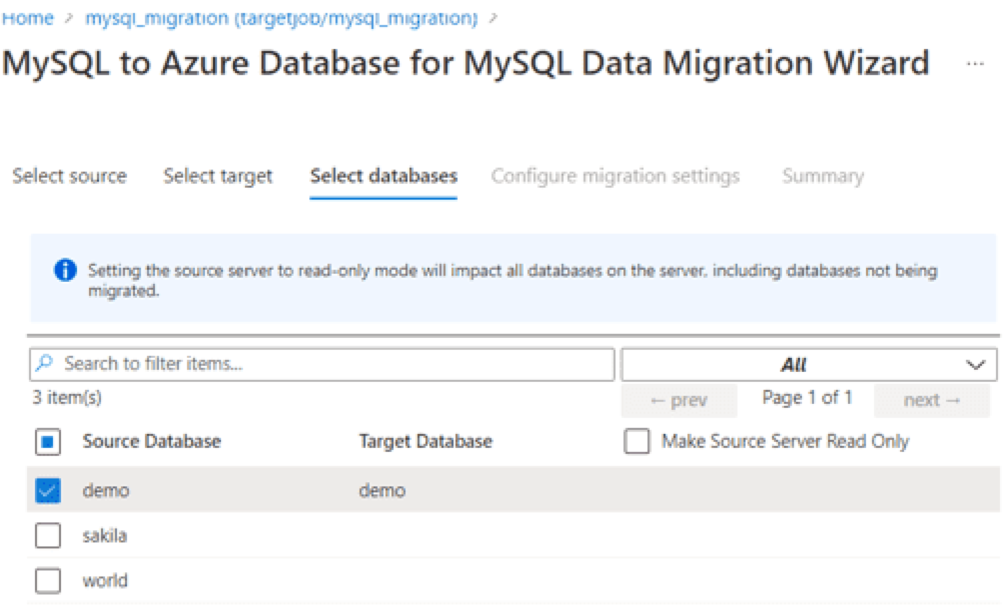
MySQL to Azure Database for MySQL Data Migration Wizard

Target server name: [redacted].mysql.database.azure.com

User Name: [redacted]

Password: ••••••••••••••••

15. Select Save, and then on the Map to target databases screen, map the source and the target database for migration.
    If the target database contains the same database name as the source database, the Azure Database Migration Service selects the target database by default.


Home > mysql_migration (targetjob/mysql_migration) >
...
Select source
Select target
Select databases
Configure migration settings
Summary
All
← prev
Page 1 of 1
next →

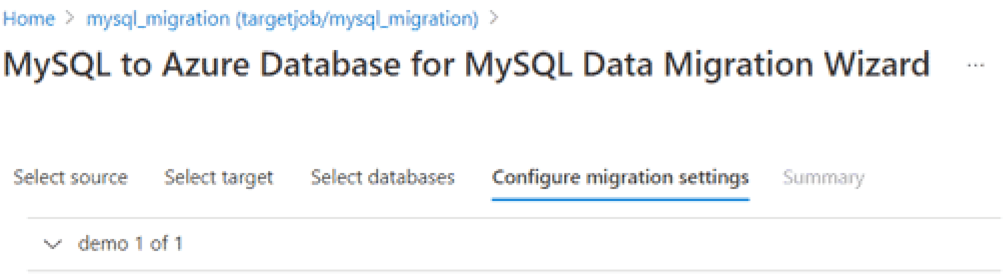
MySQL to Azure Database for MySQL Data Migration Wizard

Setting the source server to read-only mode will impact all databases on the server, including databases not being migrated.

Search to filter items...

3 item(s)

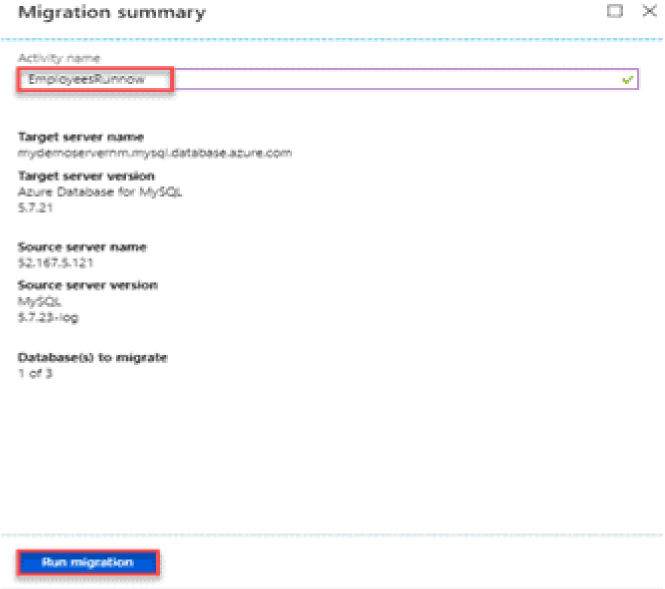
| ■ Source Database | Target Database | □ Make Source Server Read Only |
|---|---|---|
| ☑ demo | demo | |
| □ sakila | | |
| □ world | | |


Home > mysql_migration (targetjob/mysql_migration) >
...
Select source
Select target
Select databases
Configure migration settings
Summary

MySQL to Azure Database for MySQL Data Migration Wizard

demo 1 of 1

16. Select Save, on the Migration summary screen, in the Activity name text box, specify a name for the migration activity, and then review the summary to ensure that the source and target details match what you previously specified. Select **Run migration.**



Migration summary

Activity name: EmployeesRunnow

Target server name
mydemoservernm.mysql.database.azure.com

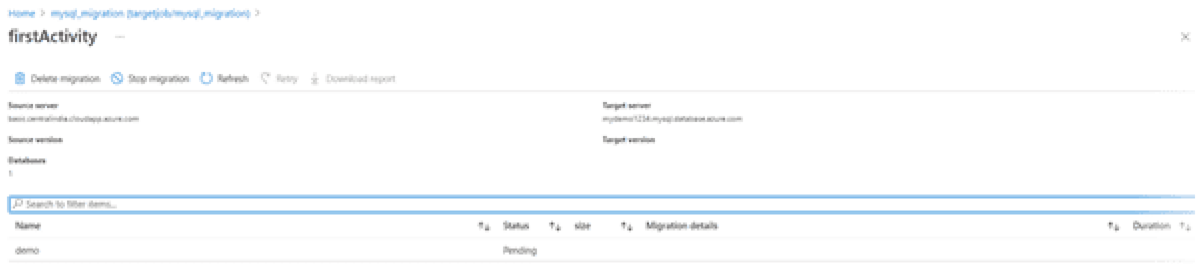
Target server version
Azure Database for MySQL
5.7.21

Source server name
52.167.5.121

Source server version
MySQL
5.7.23-log

Database(s) to migrate
1 of 3

17. The migration activity window appears, and the Status of the activity is pending. Refresh the page in sometime.



Home > mysql_migration (targetjob/mysql_migration) >
firstActivity
Delete migration
Stop migration
Refresh
Retry
Download report
Search to filter items...

Source server
basic.centralindia.cloudapp.azure.com

Target server
mydemo1234.mysql.database.azure.com

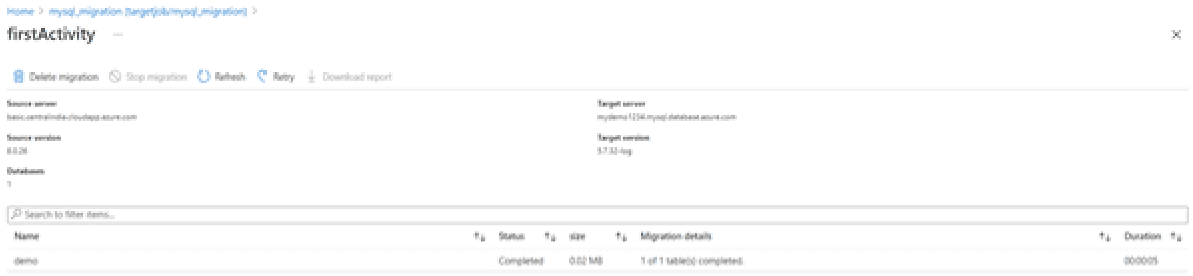
Source version

Target version

Databases
1

| Name | Status | size | Migration details | Duration |
|---|---|---|---|---|
| demo | Pending | | | |

18. The migration process is complete.



Home > mysql_migration (targetjob/mysql_migration) >
firstActivity
Delete migration
Stop migration
Refresh
Retry
Download report
Search to filter items...

Source server
basic.centralindia.cloudapp.azure.com

Target server
mydemo1234.mysql.database.azure.com

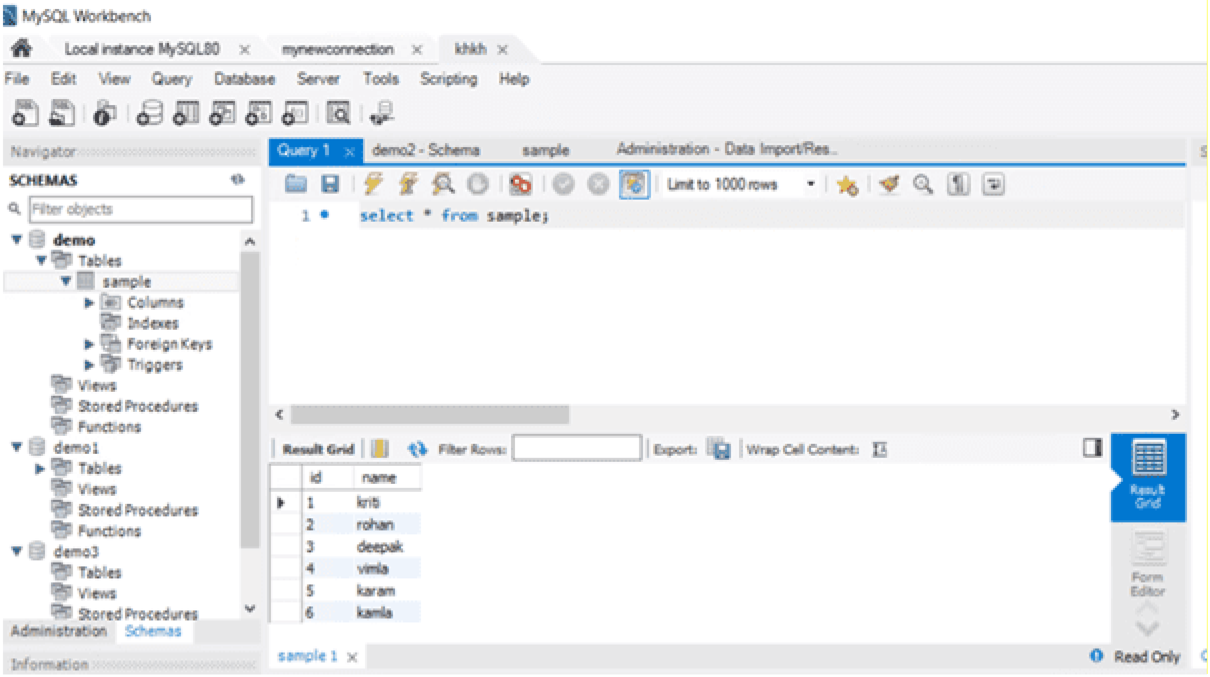
Source version
8.0.26

Target version
5.7.32-log

Databases
1

| Name | Status | size | Migration details | Duration |
|---|---|---|---|---|
| demo | Completed | 0.02 MB | 1 of 1 table(s) completed. | 00:00:05 |

19. Now check Azure database for MySQL on workbench if the data is migrated or not.



```
select * from sample;
```

| id | name |
|---|---|
| 1 | kriti |
| 2 | rohan |
| 3 | deepak |
| 4 | vimla |
| 5 | karam |
| 6 | kamla |

20. The data is successfully migrated.

## Challenges Faced:

Some major challenges that we faced were with the right version of the MySQL workbench. As Both on-premises MySQL and Azure database for MySQL version should be same. And the second challenge which we were facing was some Collation error – “unknown Collation – ‘utf8mb4_0900_ai_ci’, So this was the main challenge where we were stuck before migrating the schema of the database. We tackle this issue by replacing ‘utf8mb4_0900_ai_ci’ with ‘utf8mb4_unicode_ci’.

**Business Benefits:**

Our client wanted to migration their whole data from there on-premises environment to minimise the storage cost. This POC helped us in migrating Data from On-premises MySQL to Azure Database for MySQL in production environment with minimal downtime and budget friendly cost and helped us meeting the deadline.

By,

Kriti Bhardwaj (Data Engineer)