AZURE DATABASE FOR MYSQL

MySQL is a popular free relational database management system that Azure supports. This write-up explains how to utilize Azure Database for MySQL service for your business needs

An Azure database service that supports open-source

Contents

Introduction

MySQL is a popular relational database management system (RDBMS) that many companies and organizations use, large and small. MySQL was originally owned and sponsored by a Swedish company MySQL AB, and was later bought by Sun Microsystems in US. In 2010, Michael Widenius, the co-founder of MySQL, forked MySQL open-source project to create Marian DB when Oracle acquired Sun Microsystems. Since then, MySQL remains as free and open-source software under the terms of the GNU General Public License. Oracle owns MySQL.

MySQL runs on both Linux and Windows operating systems. MySQL distribution includes MySQL server and MySQL client. MySQL client comes in two forms. One is a command line interface (CLI) tool, simply called 'mysql'; the other is MySQL Workbench, a graphic user interface (GUI) client.

All large Cloud Service Providers (CSPs) support MySQL, including Azure, AWS, Google Cloud Platform (GCP), Oracle Cloud Services (OCS), and others. In this write-up, we focus on MySQL in Azure. We discuss how to create MySQL server and database in Azure, and how to use MySQL client running on a local workstation outside Azure to connect to and use MySQL server in Azure cloud.

Where MySQL fits in Azure

Azure offers more than 100 services. Those services fall under various categories, of which 10 are most commonly used. One such a category is Database.

Most commonly used service categories

1. Compute
2. Networking
3. Storage
4. Mobile
5. **Databases**
6. Web
7. Internet of Things (IoT)
8. Big data
9. AI
10. DevOps

In Database category, Azure offers several database services. MySQL is one of them

Databases

1. Azure Cosmos DB
2. Azure SQL Database
3. **Azure Database for MySQL**
4. Azure Database fir PostgreSQL
5. SQL Server on Azure Virtual Machine
6. Azure Synapse Analytics
7. Azure Database Migration Service
8. Azure Cache for Redis
9. Azure Database for MariaDB

Prerequisites

Before you can use MySQL in Azure, you have to have a few things in place to go about with. Those items are

1. Azure account
2. Azure subscription
3. MySQL client downloaded and installed on your workstation, including both Workbench and mysql
4. Azure CLI installed on your workstation
5. Resource Group created in your Azure account for MySQL to reside in

If you don't already have an Azure account, you can create a free-tier account. Credit card is required. The free-tier account offers $200 credit for you to use for one month. Your free-tier account comes with a free-tier subscription with limited active time. If you created and used a free-tier account before but you didn't use it for a while, chances are your free tier subscription with the account expired. If that is the case, you need to sign in your account and re-activate your subscription or purchase a new subscription in your account. Without an active subscription, you can do nothing in your account in Azure.

In addition to having an account with a working subscription, you need to install MySQL client and Azure CLI in your workstation, which are explained in detail in the following sections.

MySQL Client

MySQL client is a piece of software for you to connect to MySQL server. Without the client, you cannot create, use, and manage database(s) running on MySQL server. MySQL client comes in two forms. One is in command line interface (CLI), named as mysql; the other, in GUI, called MySQL Workbench. MySQL CLI mysql is pre-installed in Azure Cloud Shell. MySQL Workbench is not available anywhere in Azure Cloud so you need to install it on your workstation in order to use it. Though Workbench can be installed and run on a virtual machine (VM) in Azure for you to use, and the VM can be of either Linux or Windows, in this write-up, we will not discuss about that approach as it is not a common practice.

Download MySQL Client

To download MySQL Workbench, go to the link below.
https://dev.mysql.com/downloads/workbench/

Your download results in a MSI installer with a file name like
mysql-workbench-community-8.0.22-winx64.msi

To install Workbench, simply run the installer, and follow the prompts till completion. Note that the command line client 'mysql' resides in the folder you chose to install Workbench. The executable file name and path is something like this
"C:\Program Files\MySQL\MySQL Workbench 8.0 CE\mysql.exe"

Workbench

To run Workbench, open Windows Start, locate MySQL folder, click on Workbench icon, something like this
MySQL Workbench 8.0 CE

For easy access to Workbench later on, you may pin it to Start, and/or pin it to Task Bar

MySQL CLI

To run mysql command line client, open a Command Prompt window, and type mysql, like this

```
C:\Program Files\MySQL\MySQL Workbench 8.0 CE>mysql --version
mysql  Ver 8.0.22 for Win64 on x86_64 (MySQL Community Server - GPL)
```

or

C:\Program Files\MySQL\MySQL Workbench 8.0 CE>**mysql --help | more**. You will see in response the output like this below

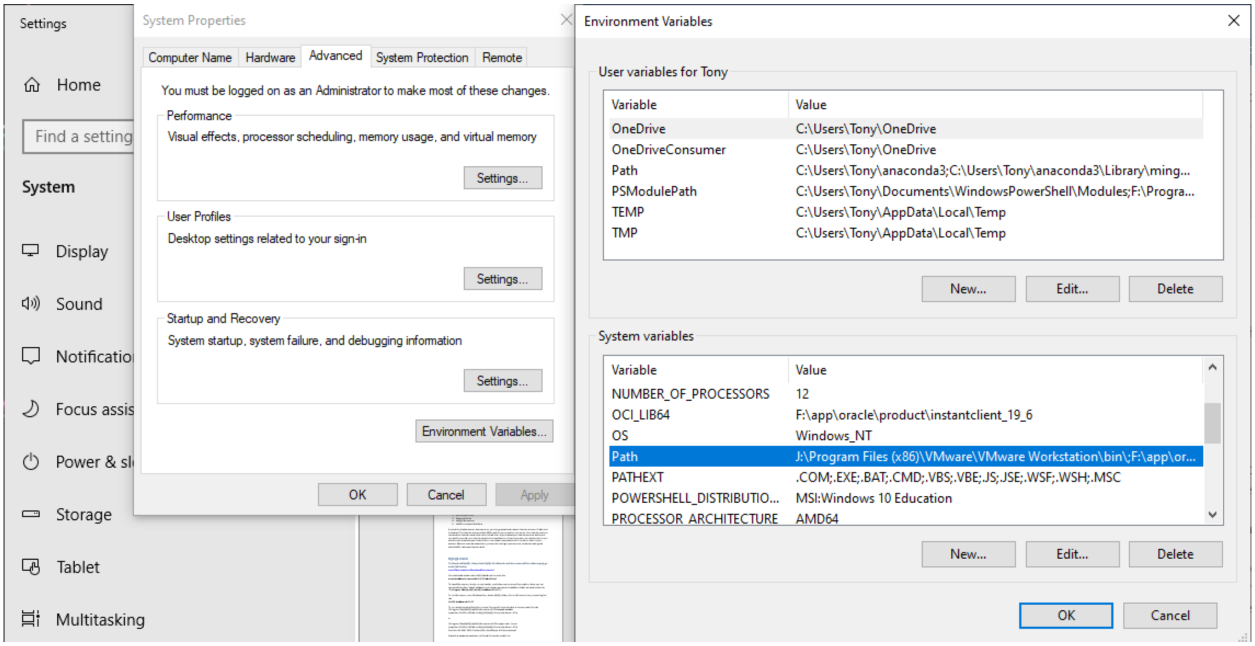
```
mysql  Ver 8.0.22 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to use
                      'rehash' to get table and field completion, but startup
                      and reconnecting may take a longer time. Disable with
                      --disable-auto-rehash.
                      (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash
                      No automatic rehashing. One has to use 'rehash' to get
                      table and field completion. This gives a quicker start of
                      mysql and disables rehashing on reconnect.
…
```

If you see an error indicating mysql not found like this below,

```
C:\Users\Tony>mysql
'mysql' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Tony>
```

you need to add Workbench installation folder to your system variable **Path** for it to be found.

To add Workbench folder to Path, open Control Panel, go to Advanced Systems Settings, click on Environment variables, find System variables, scroll down to 'Path', select it and click on Edit



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

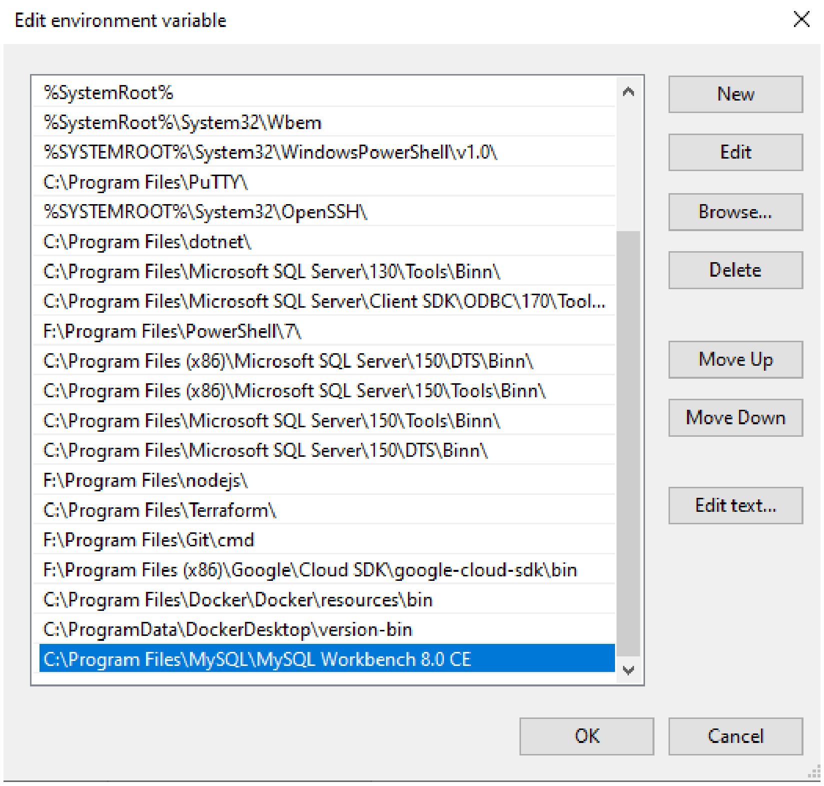
User variables for Tony

Variable	Value
OneDrive	C:\Users\Tony\OneDrive
OneDriveConsumer	C:\Users\Tony\OneDrive
Path	C:\Users\Tony\anaconda3;C:\Users\Tony\anaconda3\Library\ming...
PSModulePath	C:\Users\Tony\Documents\WindowsPowerShell\Modules;F:\Progra...
TEMP	C:\Users\Tony\AppData\Local\Temp
TMP	C:\Users\Tony\AppData\Local\Temp

System variables

Variable	Value
NUMBER_OF_PROCESSORS	12
OCI_LIB64	F:\app\oracle\product\instantclient_19_6
OS	Windows_NT
Path	J:\Program Files (x86)\VMware\VMware Workstation\bin\;F:\app\or...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
POWERSHELL_DISTRIBUTIO...	MSI:Windows 10 Education
PROCESSOR_ARCHITECTURE	AMD64

Click New, enter your Workbench installation folder path, and click OK.



```
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files\PuTTY\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\dotnet\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tool...
F:\Program Files\PowerShell\7\
C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\DTS\Binn\
F:\Program Files\nodejs\
C:\Program Files\Terraform\
F:\Program Files\Git\cmd
F:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin
C:\Program Files\Docker\Docker\resources\bin
C:\ProgramData\DockerDesktop\version-bin
C:\Program Files\MySQL\MySQL Workbench 8.0 CE
```

After you added Workbench folder to System variable Path, click OK, and click OK again to close Control Panel System window.

For your change to take effect, close your command prompt window, open a new command prompt window, and type mysql –version again, like this below, to confirm it is working

```
C:\Users\Tony>mysql --version
mysql  Ver 8.0.22 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\Tony>
```

Azure CLI

Azure CLI is required to practice creating and running MySQL in Azure Cloud to be discussed in this write-up.

To download and install Azure CLI, please go the link below
https://docs.microsoft.com/en-us/cli/azure/

If you chose to install Azure CLI on Windows, please verify that it is functioning after you completed its installation by running a simple version check command below.

```
C:\Users\Tony>az --version
azure-cli                         2.3.1

command-modules-nspkg             2.0.3
core                              2.3.1
nspkg                             3.0.4
telemetry                         1.0.4

Python location 'C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\Tony\.azure\cliextensions'

Python (Windows) 3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 02:47:15) [MSC v.1900 32 bit (Intel)]

Legal docs and information: aka.ms/AzureCliLegal

Unable to check if your CLI is up-to-date. Check your internet connection.

Please let us know how we are doing: https://aka.ms/clihats

C:\Users\Tony>
```

If your command yields either “az not found” or any other errors, you need to fix it until it works by referencing to Azure CLI documentation at the link above.

Before using Azure CLI, you need to sign in your account. Type the command below.

```
C:\Users\Tony>az login
```

A browser window pops up for you to choose an Azure account of yours to sign in. After a successful authentication with your chosen account, you see your account information displayed in your Command Prompt window like this below

```
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "604c1504-c6a3-4080-81aa-b33091104187",
    "id": "fc3a3623-e9b1-40f9-ae0b-b52678c22c26",
    "isDefault": false,
    "managedByTenants": [],
    "name": "Azure subscription 1",
    "state": "Enabled",
    "tenantId": "604c1504-c6a3-4080-81aa-b33091104187",
    "user": {
      "name": "admin-user-02@dclpearland2020.onmicrosoft.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "52a2406d-0545-429f-bbdd-9e3f9c6a6cdf",
    "id": "6d6c2372-faed-4d8f-86d1-3019ed20ea44",
    "isDefault": true,
    "managedByTenants": [],
    "name": "DCL Pearland 2020",
    "state": "Enabled",
    "tenantId": "52a2406d-0545-429f-bbdd-9e3f9c6a6cdf",
    "user": {
      "name": "admin-user-02@dclpearland2020.onmicrosoft.com",
      "type": "user"
    }
  }
]

C:\Users\Tony>
```

To check your subscriptions in your account, run the command below

```
C:\Users\Tony>az account list --query "[].{name:name}" --output table
Name
-----------------------------
MAE ITAA Customer Service Dev
Azure subscription 1
DCL Pearland 2020
```

```
C:\Users\Tony>
```

If you see more than one subscription like what was shown above, set one active subscription as your default subscription using the example command below.

```
C:\Users\Tony>az account set --subscription "DCL Pearland 2020"

C:\Users\Tony>
```

If you see only one subscription like this below, the subscription is already your default subscription. No need to set it to be your default subscription again.

```
C:\Users\Tony>az account list --query "[].{name:name}" --output table
Name
-----------------------------
Azure subscription 1

C:\Users\Tony>
```

To verify that your Azure CLI is accessing your account with your chosen subscription properly, run the command below to list existing resource groups in your account.

```
C:\Users\Tony>az group list --query "[].{name:name,location:location}" --output table
```

You should see an output like this below.

```
Name                                  Location
----------------------------------    --------------
NetworkWatcherRG                      southcentralus
DefaultResourceGroup-SCUS             southcentralus
cloud-shell-storage-southcentralus    southcentralus
VstsRG-dclpearland2020-befd           centralus

C:\Users\Tony>
```

If you see only an empty listing, it indicates that you have not set up any Resource Group (RG) in your account yet. You need to set up at least one RG for your MySQL to use in Azure.

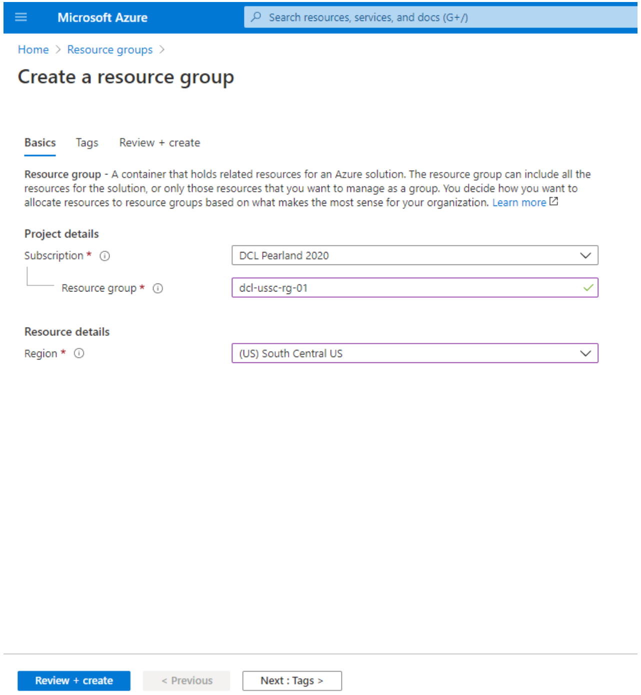
Resource Group

Resource Group (RG) is a container that groups your cloud resources together, including databases and resources of other services. You can create multiple RGs to organize your resources by location, by projects, or by any other denomination that makes sense to you. You can create a resource group using either Azure Portal or Azure CLI.

Creating RG using Portal

To create RG using Portal, sign in your account, select Resource Groups, click on Add, specify the subscription to use, RG name, and location, like the example below. Once specified to your liking, you may either click Review + Create or Next: Tags >, like screen shots below

RG to create without a tag



Create a resource group

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

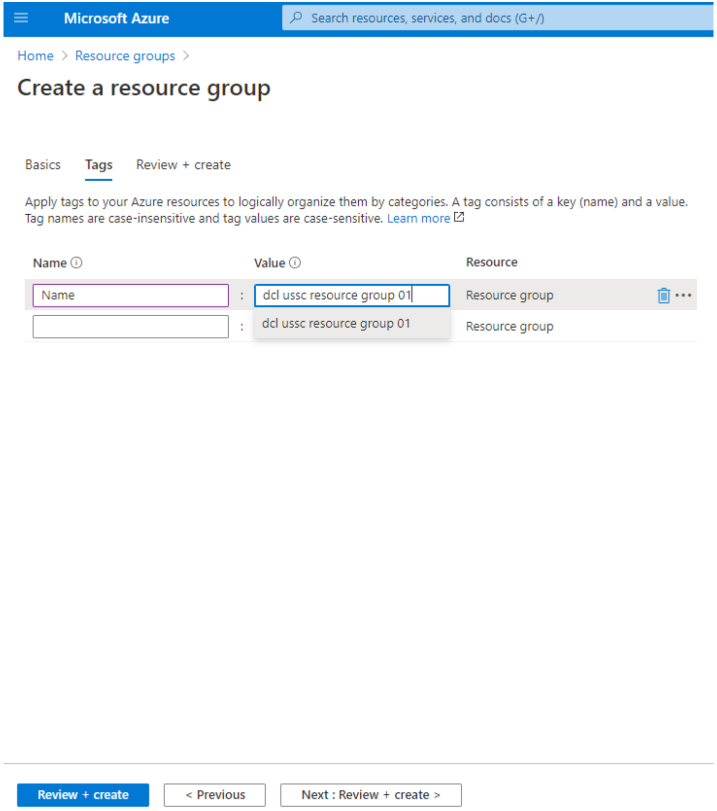
Subscription *: DCL Pearland 2020

Resource group *: dcl-ussc-rg-01

Resource details

Region *: (US) South Central US

RG to create with a name tag

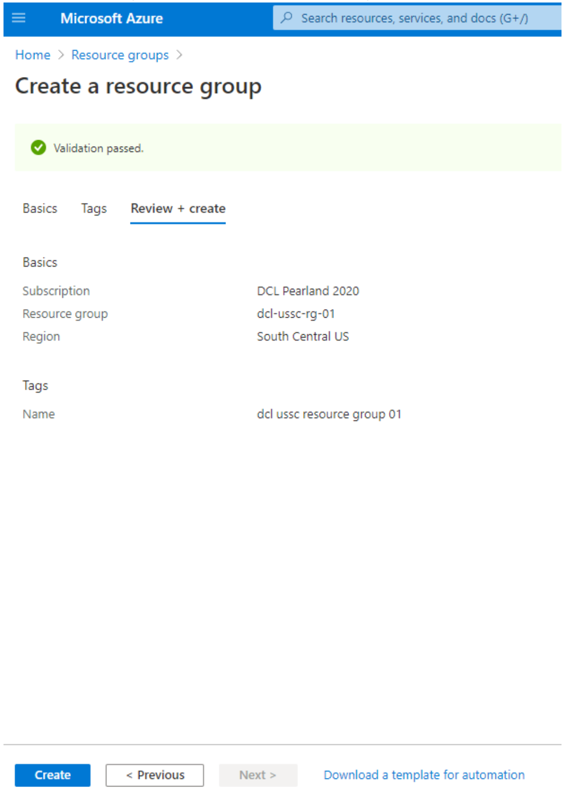


Create a resource group

Apply tags to your Azure resources to logically organize them by categories. A tag consists of a key (name) and a value. Tag names are case-insensitive and tag values are case-sensitive. Learn more

Name		Value	Resource
Name	:	dcl ussc resource group 01	Resource group
	:	dcl ussc resource group 01	Resource group

Azure lets you to review once you clicked on Review + Create.



Create a resource group

Validation passed.

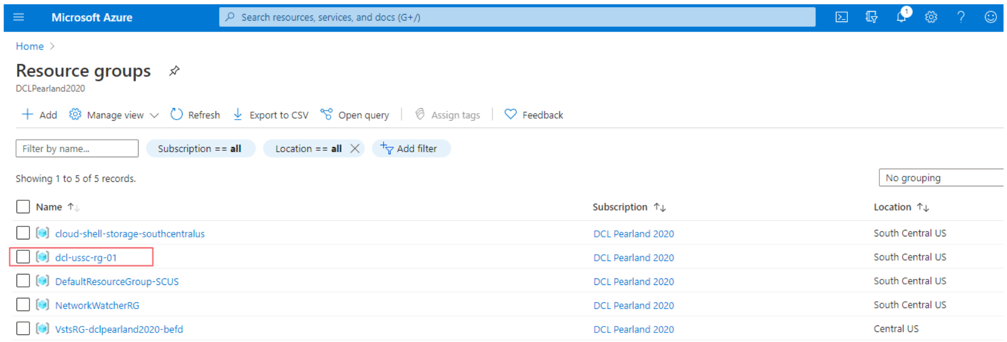
Basics

Subscription	DCL Pearland 2020
Resource group	dcl-ussc-rg-01
Region	South Central US

Tags

Name	dcl ussc resource group 01

After review, you click on Create, and it results in something like this below, indicating the resource group was created and it is available for use



Resource groups
DCLPearland2020

Showing 1 to 5 of 5 records.

Name	Subscription	Location
cloud-shell-storage-southcentralus	DCL Pearland 2020	South Central US
dcl-ussc-rg-01	DCL Pearland 2020	South Central US
DefaultResourceGroup-SCUS	DCL Pearland 2020	South Central US
NetworkWatcherRG	DCL Pearland 2020	South Central US
VstsRG-dclpearland2020-befd	DCL Pearland 2020	Central US

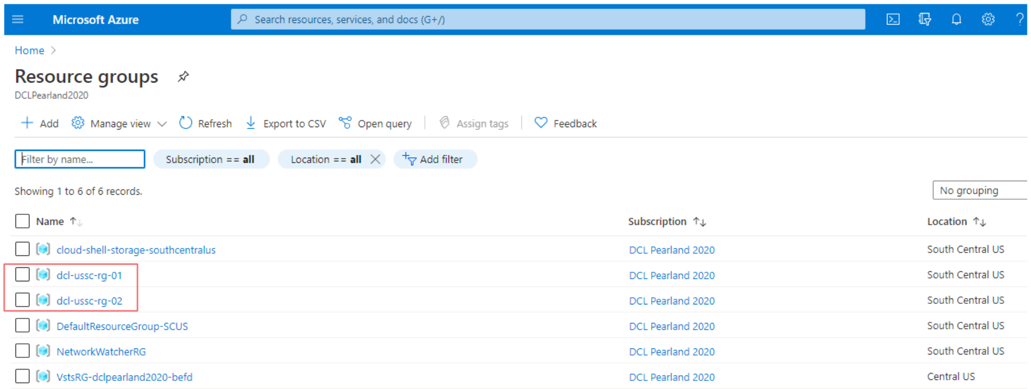
Creating RG using CLI

At Command Prompt, type the command below to create a resource group
C:\Users\Tony>**az group create --location southcentralus --name dcl-ussc-rg-02**

Your command returns an output in JSON format as shown below

```
{
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg-02",
  "location": "southcentralus",
  "managedBy": null,
  "name": "dcl-ussc-rg-02",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}

C:\Users\Tony>
```

Thus, you have created two RGs. One was created using Portal; the other, using CLI. Either one can be used for MySQL



Showing 1 to 6 of 6 records.

Name	Subscription	Location
cloud-shell-storage-southcentralus	DCL Pearland 2020	South Central US
dcl-ussc-rg-01	DCL Pearland 2020	South Central US
dcl-ussc-rg-02	DCL Pearland 2020	South Central US
DefaultResourceGroup-SCUS	DCL Pearland 2020	South Central US
NetworkWatcherRG	DCL Pearland 2020	South Central US
VstsRG-dclpearland2020-befd	DCL Pearland 2020	Central US

To list your resource groups using CLI, type the command below.

```
C:\Users\Tony>az group list --query "[].{name:name,location:location}" --output table
Name                                  Location
----------------------------------    --------------
NetworkWatcherRG                      southcentralus
DefaultResourceGroup-SCUS             southcentralus
cloud-shell-storage-southcentralus    southcentralus
dcl-ussc-rg-01                        southcentralus
dcl-ussc-rg-02                        southcentralus
VstsRG-dclpearland2020-befd           centralus

C:\Users\Tony>
```

Now that you have taken care of all prerequisites, you are ready to create MySQL instances in Azure

MySQL in Azure

Before creating MySQL instances in Azure, be aware of the following:

You have two MySQL server Choices

- Single server
- Flexible server – available but will fail to create in a free-tier account and subscription

You have three workload types

- Prod (Small / Medium)
- Prod (Large) – available but will fail to create in a free-tier account and subscription
- Dev

If you work with a free-tier subscription, it is recommended that you choose **Singer server** and **Dev** workload type.

Naming Standard

Before creating MySQL instances in Azure, it is strongly recommended that you have a naming standard (NS) in place to go by in naming your resources properly. In reality and for all practical purposes, it matters whether you have a good naming standard or not that governs how you name your resources to be created and changed moving forward. Without it, unnecessary name confusions will certainly happen when your cloud environment keeps growing overtime with more and more objects come into existence. It affects your ability to manage. With a good NS, it helps you a great deal in identifying what is what for, what is owned by whom, and where it belongs to. It helps monitoring, trouble-shooting, and automation, to say the least.

In this write-up, a location and service-based NS is used in naming our MySQL related cloud objects.

By this NS, an object will have a name that includes at least three components. One indicates the stakeholder who owns it; the second, location; and the third, service.

For instance, for a resource group to host MySQL instance, its name is something like this below
dcl-ussc-rg-01
where “dcl” is the stakeholder, which can be a business name, department name, or a project name; “ussc”, which stands for South Central US; “rg”, abbreviated for resource group service; and last but not least, “01”, as a unique ID that distinguishes this object from other similar objects should they come online in future.

Here below are a few MySQL object names we will use in this write-up

Object name	Description	Comment
dcl-ussc-rg-01	Resource Group in South Central US region	
dcl-ussc-rg-02	Resource Group in South Central US region	
dcl-ussc-db-mysql-01	MySQL server	
dcl-ussc-db-mysql-02	MySQL server	
dcl-ussc-db-mysql-01-db-01	MySQL database	
dcl-ussc-db-mysql-02-db-02	MySQL database	

MySQL server and MySQL database

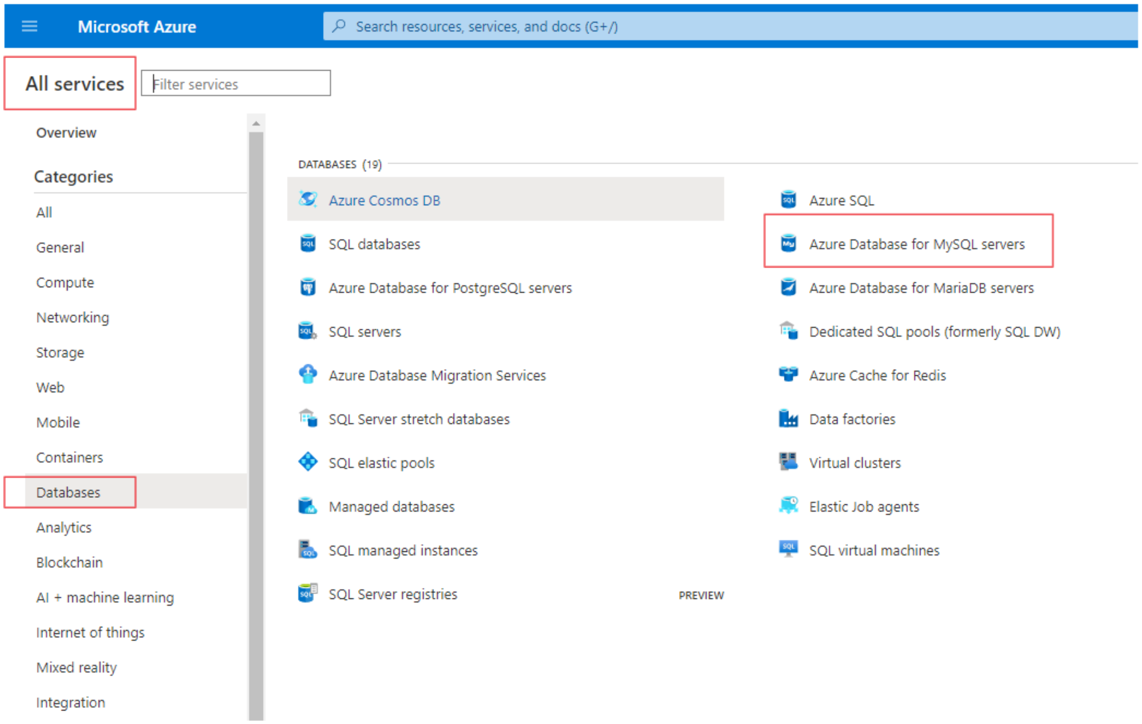
MySQL server is an instance that runs MySQL databases on it.

A MySQL server instance, upon creation, comes with a default database called 'mysql'. Best practice is such that DBA and/or developers will create a non-default database to store and use their data, which is more customized for their needs and with added security. A MySQL server instance is capable of having and running multiple databases simultaneously. Using Azure Portal, however, you can only create MySQL server instance, and not databases in that instance. To create a new database in the server instance, you need to use MySQL client, either Workbench or CLI mysql. Workbench has to be running on your local workstation or on a VM in Azure; CLI mysql can be running on your local workstation or in Azure Cloud Shell, pre-installed for you to use in there.

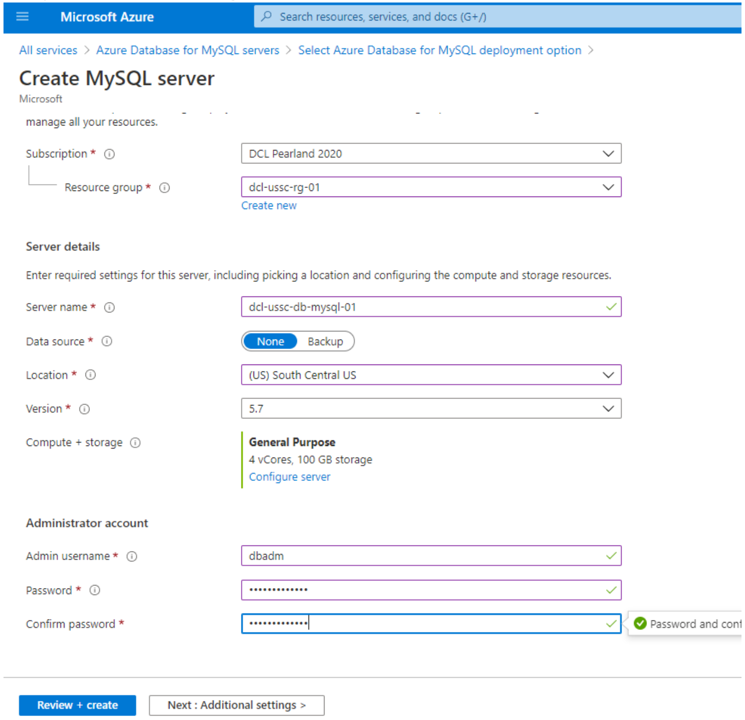
Creating MySQL server using Portal

To create a MySQL server instance in Azure Portal, follow the five (5) steps below.

Step 1 – Sign in -> All services -> Database -> Azure Database for MySQL servers



Step 2 – Add -> Single Server -> Dev -> Select Azure Database for MySQL deployment option


Microsoft Azure
Search resources, services, and docs (G+/)
All services > Azure Database for MySQL servers > Select Azure Database for MySQL deployment option >
Review + create
Next : Additional settings >

Create MySQL server

Microsoft

manage all your resources.

Subscription * DCL Pearland 2020

Resource group * dcl-ussc-rg-01

Create new

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * dcl-ussc-db-mysql-01

Data source * None Backup

Location * (US) South Central US

Version * 5.7

Compute + storage **General Purpose** 4 vCores, 100 GB storage Configure server

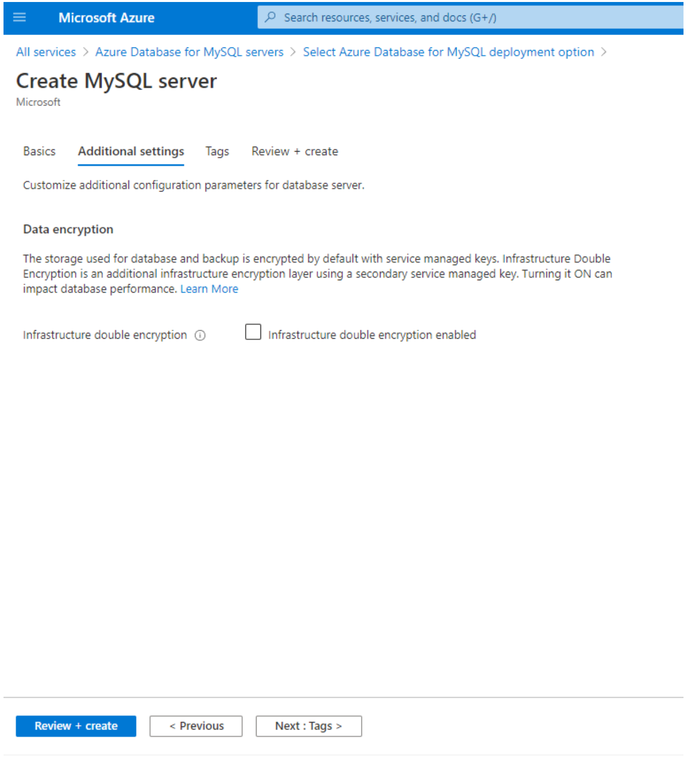
Administrator account

Admin username * dbadm

Password * •••••••••••••

Confirm password * ••••••••••••• ✓ Password and conf

Step 3 – Additional Settings (take default, leave infrastructure double encryption enabled unchecked)


Microsoft Azure
Search resources, services, and docs (G+/)
All services > Azure Database for MySQL servers > Select Azure Database for MySQL deployment option >
Basics
Additional settings
Tags
Review + create
Review + create
< Previous
Next : Tags >

Create MySQL server

Microsoft

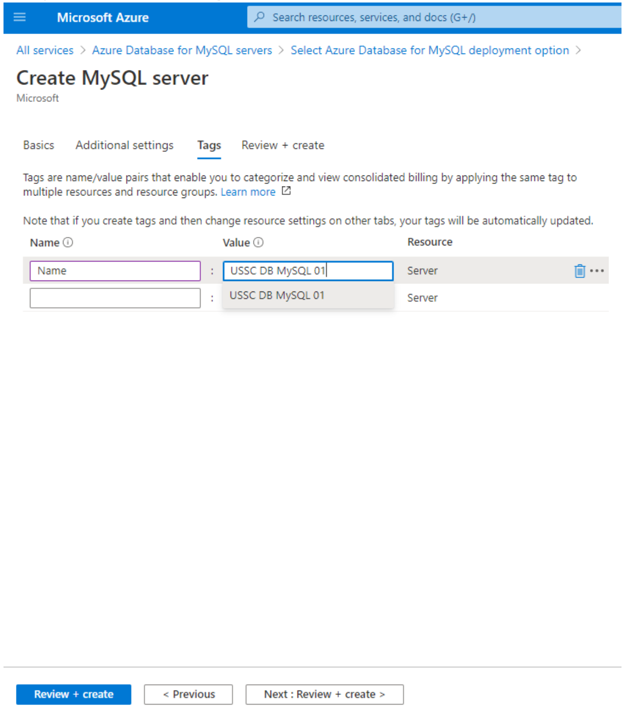
Customize additional configuration parameters for database server.

Data encryption

The storage used for database and backup is encrypted by default with service managed keys. Infrastructure Double Encryption is an additional infrastructure encryption layer using a secondary service managed key. Turning it ON can impact database performance. Learn More

Infrastructure double encryption □ Infrastructure double encryption enabled

Step 4 – Name Tag



Microsoft Azure
Search resources, services, and docs (G+/)
All services > Azure Database for MySQL servers > Select Azure Database for MySQL deployment option >

Create MySQL server

Microsoft

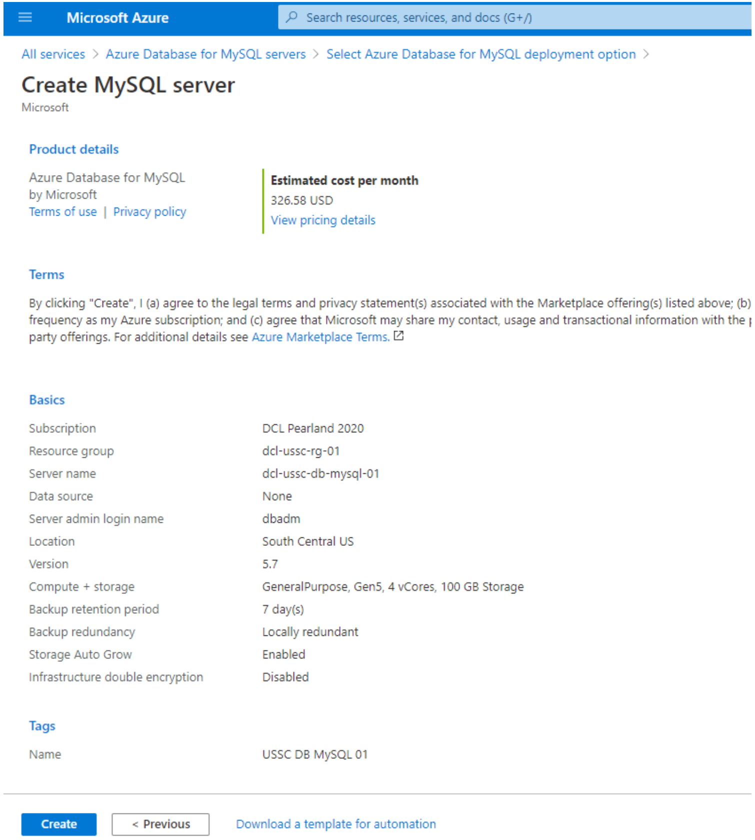
Basics
Additional settings
Tags
Review + create

Tags are name/value pairs that enable you to categorize and view consolidated billing by applying the same tag to multiple resources and resource groups. Learn more

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name		Value	Resource
Name	:	USSC DB MySQL 01	Server
	:	USSC DB MySQL 01	Server

Step 5 – Review and Create



Microsoft Azure
Search resources, services, and docs (G+/)
All services > Azure Database for MySQL servers > Select Azure Database for MySQL deployment option >

Create MySQL server

Microsoft

Product details

Azure Database for MySQL
by Microsoft
Terms of use | Privacy policy

Estimated cost per month
326.58 USD
View pricing details

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the party offerings. For additional details see Azure Marketplace Terms.

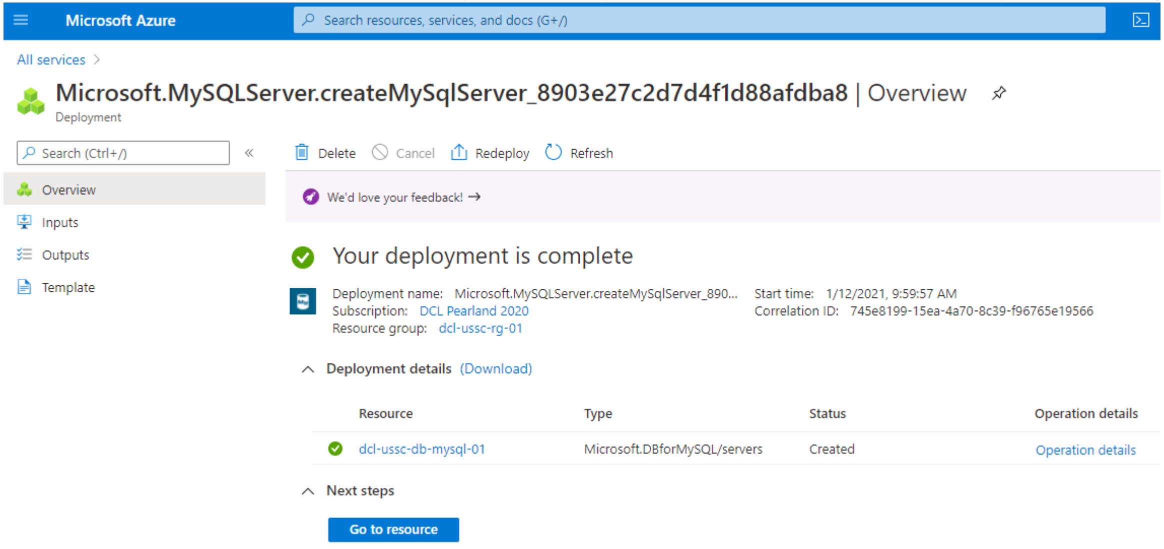
Basics

Subscription	DCL Pearland 2020
Resource group	dcl-ussc-rg-01
Server name	dcl-ussc-db-mysql-01
Data source	None
Server admin login name	dbadm
Location	South Central US
Version	5.7
Compute + storage	GeneralPurpose, Gen5, 4 vCores, 100 GB Storage
Backup retention period	7 day(s)
Backup redundancy	Locally redundant
Storage Auto Grow	Enabled
Infrastructure double encryption	Disabled

Tags

Name	USSC DB MySQL 01

Click on Create. A few moments later, your first MySQL server instance is created, up and running



Microsoft.MySQLServer.createMySqlServer_8903e27c2d7d4f1d88afdba8 | Overview

Deployment

Your deployment is complete

Deployment name: Microsoft.MySQLServer.createMySqlServer_890...
Subscription: DCL Pearland 2020
Resource group: dcl-ussc-rg-01
Start time: 1/12/2021, 9:59:57 AM
Correlation ID: 745e8199-15ea-4a70-8c39-f96765e19566

Deployment details (Download)

Resource	Type	Status	Operation details
dcl-ussc-db-mysql-01	Microsoft.DBforMySQL/servers	Created	Operation details

Next steps

Creating MySQL server and database using CLI

As we mentioned before, in Azure Portal, you can only create MySQL server and not MySQL database. In CLI, you can do both.

Creating MySQL server in CLI

To create a MySQL **server** in Azure, run this command below
C:\Users\Tony>**az mysql server create -g dcl-ussc-rg-02 -l southcentralus --sku-name B_Gen5_1 --public-network-access Enabled -n "dcl-ussc-db-mysql-02" --tags Name="USSC DB MySQL 02" -u dbadm -p "<your password>" –verbose**

The command takes a number of parameters, some are required, others are optional. In the example, those parameters are:

-g dcl-ussc-rg-02 (specifying a resource group; required)
-l southcentralus (specifying a location; required)
--sku-name B_Gen5_1 (specifying server resource configuration; required)
-n "dcl-ussc-db-mysql-02" (specifying the server name; required)
--tags Name="USSC DB MySQL 02" (specifying a tag; optional)
-u dbadm (specifying admin-login name; required)
-p "<password>" (specifying admin password; required)
--verbose (specifying logging level, optional)

Output in JSON returns once the command run finishes, like this below

```
{
  "administratorLogin": "dbadm",
  "byokEnforcement": "Disabled",
  "earliestRestoreDate": "2021-01-12T17:23:27.960000+00:00",
  "fullyQualifiedDomainName": "dcl-ussc-db-mysql-02.mysql.database.azure.com",
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg-02/providers/Microsoft.DBforMySQL/servers/dcl-ussc-db-mysql-02",
  "identity": null,
  "infrastructureEncryption": "Disabled",
  "location": "southcentralus",
  "masterServerId": "",
  "minimalTlsVersion": "TLSEnforcementDisabled",
  "name": "dcl-ussc-db-mysql-02",
  "privateEndpointConnections": [],
  "publicNetworkAccess": "Enabled",
  "replicaCapacity": 5,
  "replicationRole": "None",
  "resourceGroup": "dcl-ussc-rg-02",
  "sku": {
    "capacity": 1,
    "family": "Gen5",
    "name": "B_Gen5_1",
    "size": null,
    "tier": "Basic"
  },
  "sslEnforcement": "Enabled",
  "storageProfile": {
    "backupRetentionDays": 7,
    "geoRedundantBackup": "Disabled",
    "storageAutogrow": "Enabled",
    "storageMb": 5120
  },
  "tags": {
    "Name": "USSC DB MySQL 02"
  },
  "type": "Microsoft.DBforMySQL/servers",
  "userVisibleState": "Ready",
  "version": "5.7"
}
```

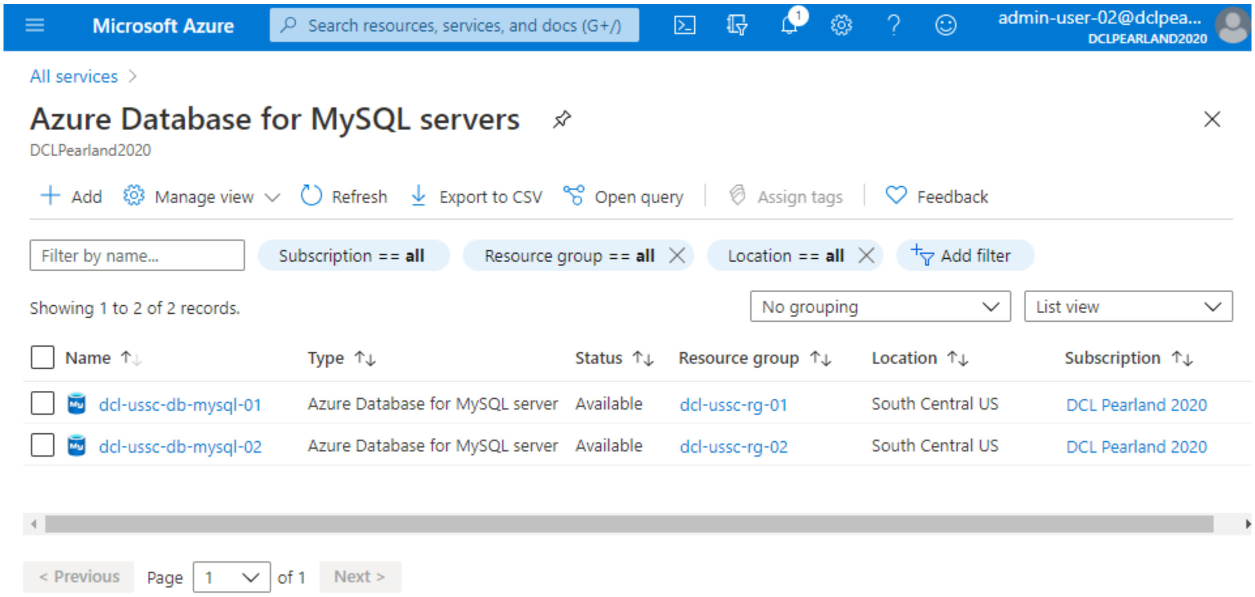
command ran in **182.832 seconds**.

```
C:\Users\Tony>
```

Using Azure CLI, the command below shows two servers that you have created

```
C:\Users\Tony>az mysql server list --query "[].{name:name}" --output tsv
dcl-ussc-db-mysql-01
dcl-ussc-db-mysql-02
C:\Users\Tony>
```

In Portal, you can also see those two MySQL servers



Azure Database for MySQL servers

DCLPearland2020

Subscription == all, Resource group == all, Location == all

Showing 1 to 2 of 2 records.

No grouping, List view

Name	Type	Status	Resource group	Location	Subscription
dcl-ussc-db-mysql-01	Azure Database for MySQL server	Available	dcl-ussc-rg-01	South Central US	DCL Pearland 2020
dcl-ussc-db-mysql-02	Azure Database for MySQL server	Available	dcl-ussc-rg-02	South Central US	DCL Pearland 2020

Creating MySQL database in CLI

To create a new database in a server, run this command below

```
C:\Users\Tony>az mysql db create -g dcl-ussc-rg-01 -s dcl-ussc-db-mysql-01 -n dcl-ussc-db-mysql-01-db-01
{
  "charset": "latin1",
  "collation": "latin1_swedish_ci",
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg-01/providers/Microsoft.DBforMySQL/servers/dcl-ussc-db-mysql-01/databases/dcl-ussc-db-mysql-01-db-01",
  "name": "dcl-ussc-db-mysql-01-db-01",
  "resourceGroup": "dcl-ussc-rg-01",
  "type": "Microsoft.DBforMySQL/servers/databases"
}

C:\Users\Tony>
```

And run this command also to create a database in another server

```
C:\Users\Tony>az mysql db create -g dcl-ussc-rg-02 -s dcl-ussc-db-mysql-02 -n dcl-ussc-db-mysql-02-db-01
{
  "charset": "latin1",
  "collation": "latin1_swedish_ci",
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg-02/providers/Microsoft.DBforMySQL/servers/dcl-ussc-db-mysql-02/databases/dcl-ussc-db-mysql-02-db-01",
  "name": "dcl-ussc-db-mysql-02-db-01",
  "resourceGroup": "dcl-ussc-rg-02",
  "type": "Microsoft.DBforMySQL/servers/databases"
}

C:\Users\Tony>
```

To list MySQL databases, run this command

```
C:\Users\Tony>az mysql db list -g dcl-ussc-rg-01 -s dcl-ussc-db-mysql-01 --query "[].{name:name}" --output tsv
information_schema
dcl-ussc-db-mysql-01-db-01
mysql
performance_schema
sys
```

and this command

```
C:\Users\Tony>az mysql db list -g dcl-ussc-rg-02 -s dcl-ussc-db-mysql-02 --query "[].{name:name}" --output tsv
information_schema
dcl-ussc-db-mysql-02-db-01
mysql
performance_schema
sys

C:\Users\Tony>
```

In Azure Portal, you don’t see databases in servers. They are not visible.

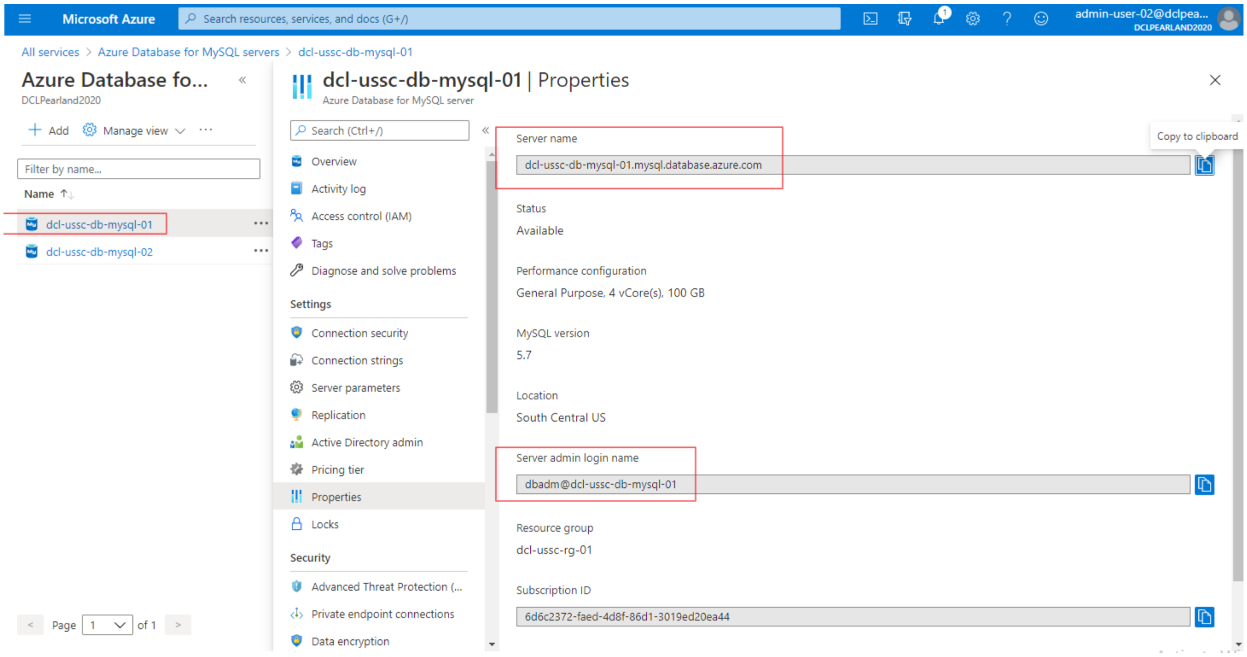
Connecting to MySQL in Azure

Before you can make any use of MySQL in Azure, you have to connect to it.

To connect to your MySQL server running in Azure cloud using your MySQL client running on your workstation, you need to know a few parameters in order to make a right connection.

1. MySQL server DNS name
2. Server admin login name
3. Server admin login password
4. Server SSL certificate

You may find server DNS name and admin login name in your MySQL server Properties.



Server name

dcl-ussc-db-mysql-01.mysql.database.azure.com

Status

Available

Performance configuration

General Purpose, 4 vCore(s), 100 GB

MySQL version

5.7

Location

South Central US

Server admin login name

dbadm@dcl-ussc-db-mysql-01

Resource group

dcl-ussc-rg-01

Subscription ID

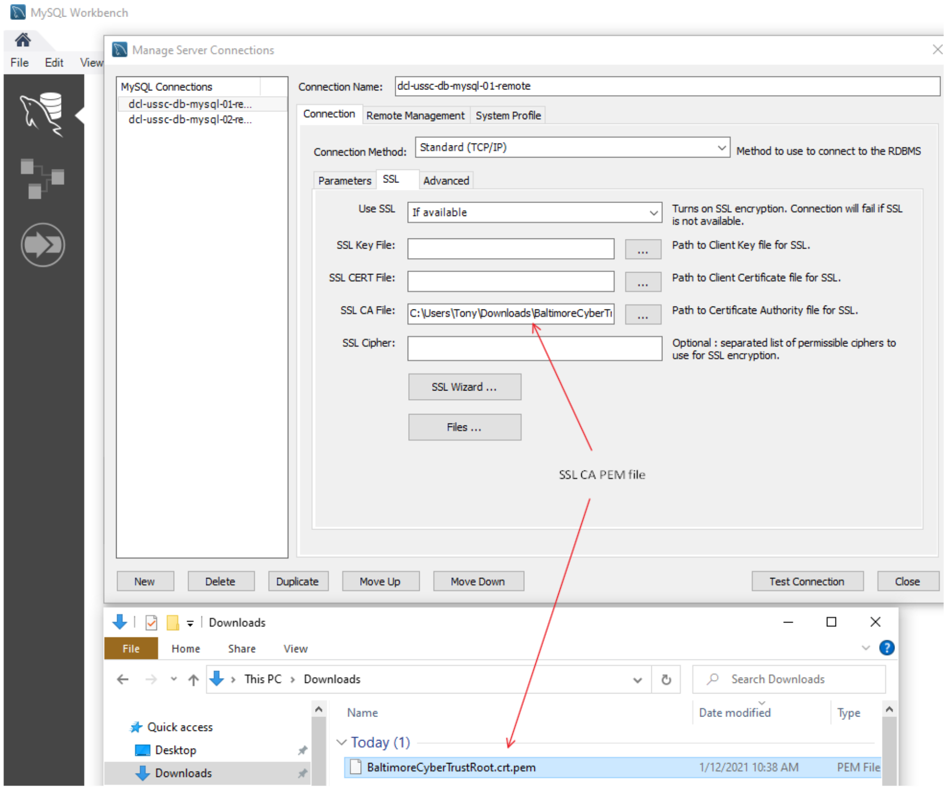
6d6c2372-faed-4d8f-86d1-3019ed20ea44

You know the admin password when you created your server. You may find and obtain SSL certificate at this link https://docs.microsoft.com/en-us/azure/mysql/howto-configure-ssl and download it to your workstation.

Now that you have acquired all the connection parameters, you are ready to connect to your server in Azure

Connecting using MySQL Workbench for Windows

Open Workbench, click on Databases, click on Manage Connections, click on New, select SSL tab, enter your downloaded SSL CA PEM file with its complete path, like this below



Manage Server Connections

MySQL Connections: dcl-ussc-db-mysql-01-re..., dcl-ussc-db-mysql-02-re...

Connection Name: dcl-ussc-db-mysql-01-remote

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Use SSL: If available — Turns on SSL encryption. Connection will fail if SSL is not available.

SSL Key File: — Path to Client Key file for SSL.

SSL CERT File: — Path to Client Certificate file for SSL.

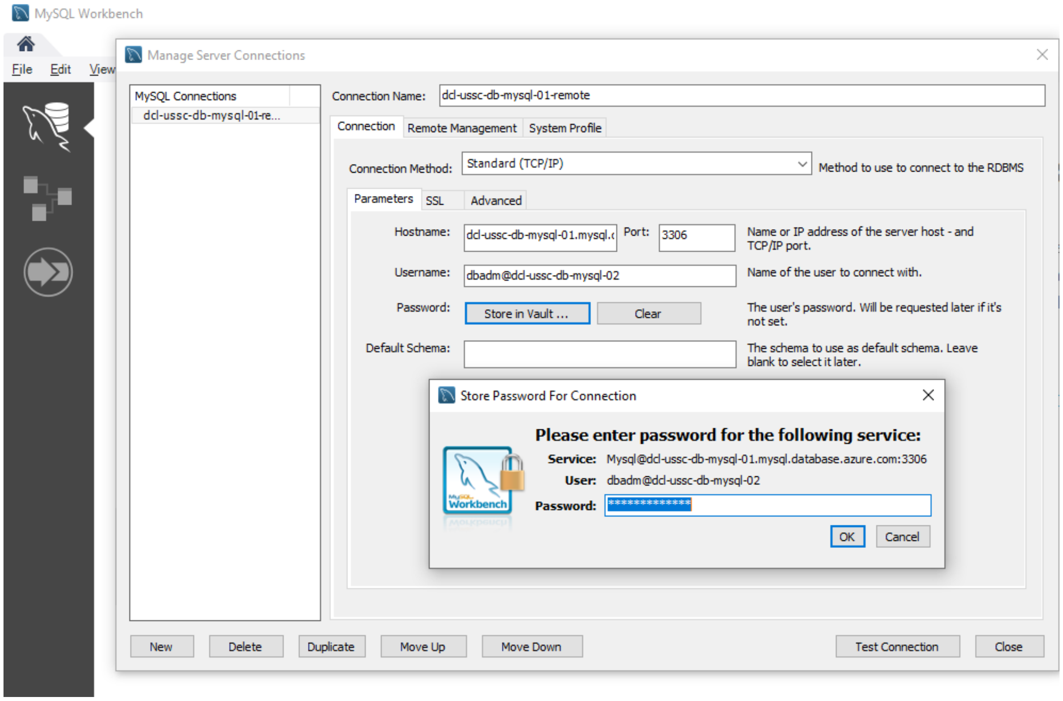
SSL CA File: C:\Users\Tony\Downloads\BaltimoreCyberT — Path to Certificate Authority file for SSL.

SSL Cipher: — Optional : separated list of permissible ciphers to use for SSL encryption.

SSL CA PEM file

Downloads: BaltimoreCyberTrustRoot.crt.pem — 1/12/2021 10:38 AM — PEM File

Select Parameters tab. Enter the required parameters like this below



Manage Server Connections

MySQL Connections: dcl-ussc-db-mysql-01-re...

Connection Name: dcl-ussc-db-mysql-01-remote

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: dcl-ussc-db-mysql-01.mysql.c Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: dbadm@dcl-ussc-db-mysql-02 — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

Store Password For Connection

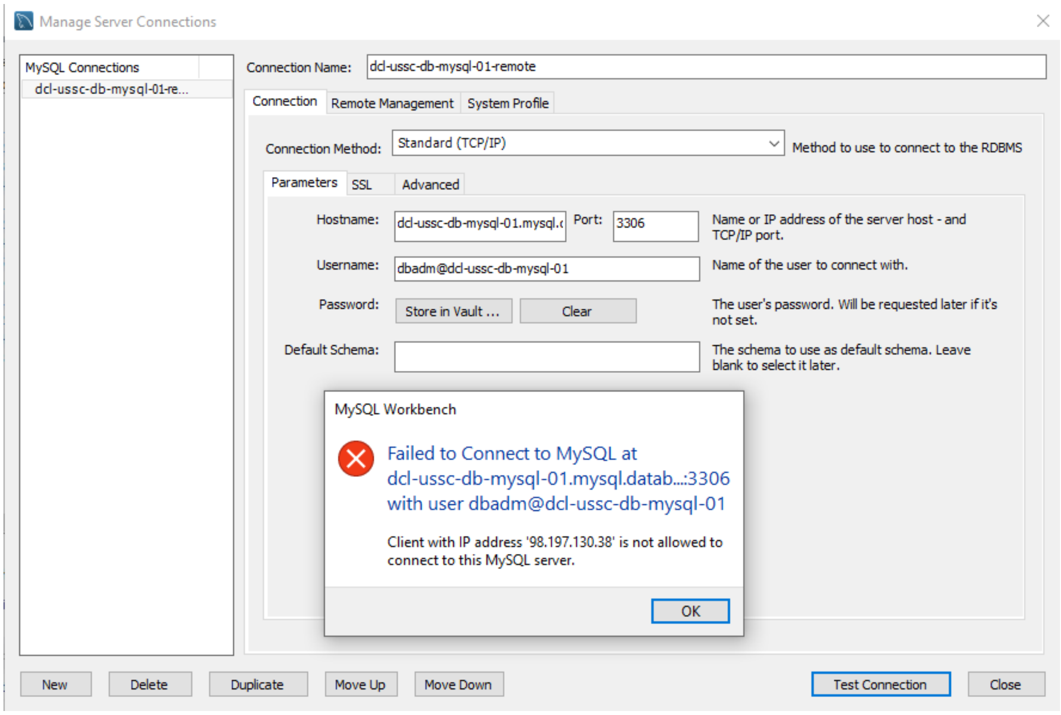
Please enter password for the following service:

Service: Mysql@dcl-ussc-db-mysql-01.mysql.database.azure.com:3306

User: dbadm@dcl-ussc-db-mysql-02

Password: *************

Click OK.
Click Test Connection



Connection Name: dcl-ussc-db-mysql-01-remote

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: dcl-ussc-db-mysql-01.mysql.c Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: dbadm@dcl-ussc-db-mysql-01 — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

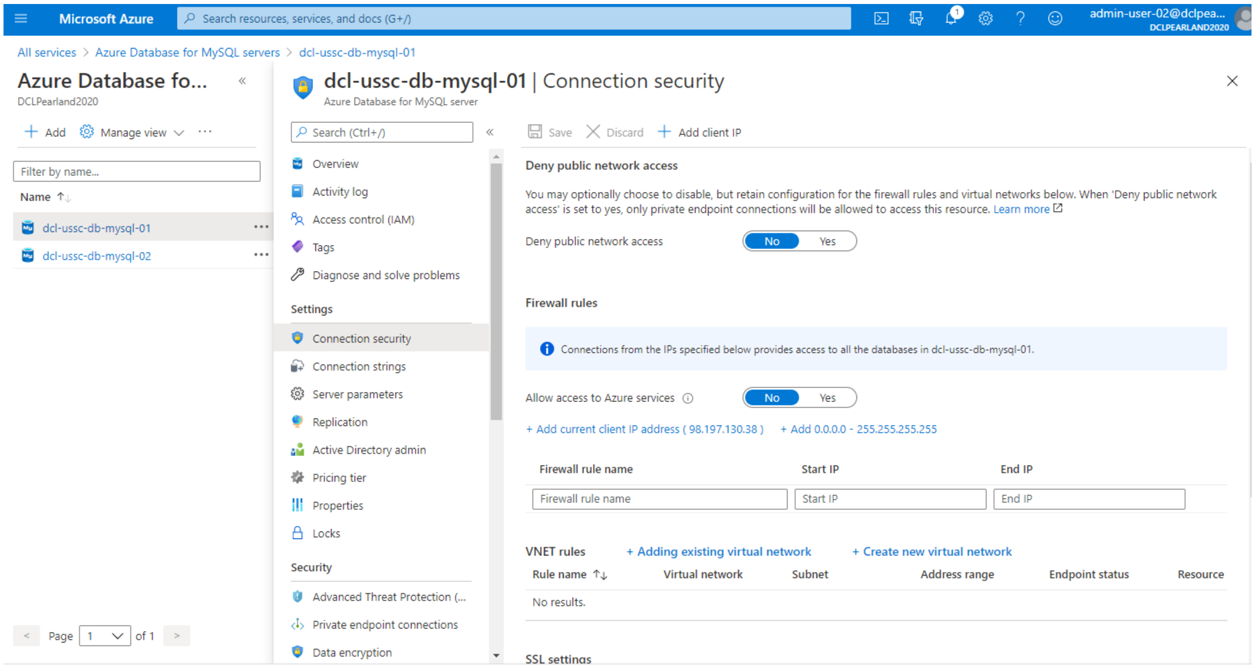
MySQL Workbench

Failed to Connect to MySQL at dcl-ussc-db-mysql-01.mysql.datab...:3306 with user dbadm@dcl-ussc-db-mysql-01

Client with IP address '98.197.130.38' is not allowed to connect to this MySQL server.

OK

If you see Failed to connect error stating "Client with IP address … is not allowed to connect to this MySQL server", The IP is your workstation IP that your Internet service provider assigned to you by DHCP, if you are working from home. If you are working in your office, it is assigned by your cooperate network system. You don't need to be overly concerned with where did this IP come from. All you need to do is simply remember this IP, go to Azure Portal, select the server, select Connection security



dcl-ussc-db-mysql-01 | Connection security

Deny public network access

You may optionally choose to disable, but retain configuration for the firewall rules and virtual networks below. When 'Deny public network access' is set to yes, only private endpoint connections will be allowed to access this resource. Learn more

Deny public network access: No [selected] Yes

Firewall rules

Connections from the IPs specified below provides access to all the databases in dcl-ussc-db-mysql-01.

Allow access to Azure services: No [selected] Yes

+ Add current client IP address (98.197.130.38) + Add 0.0.0.0 - 255.255.255.255

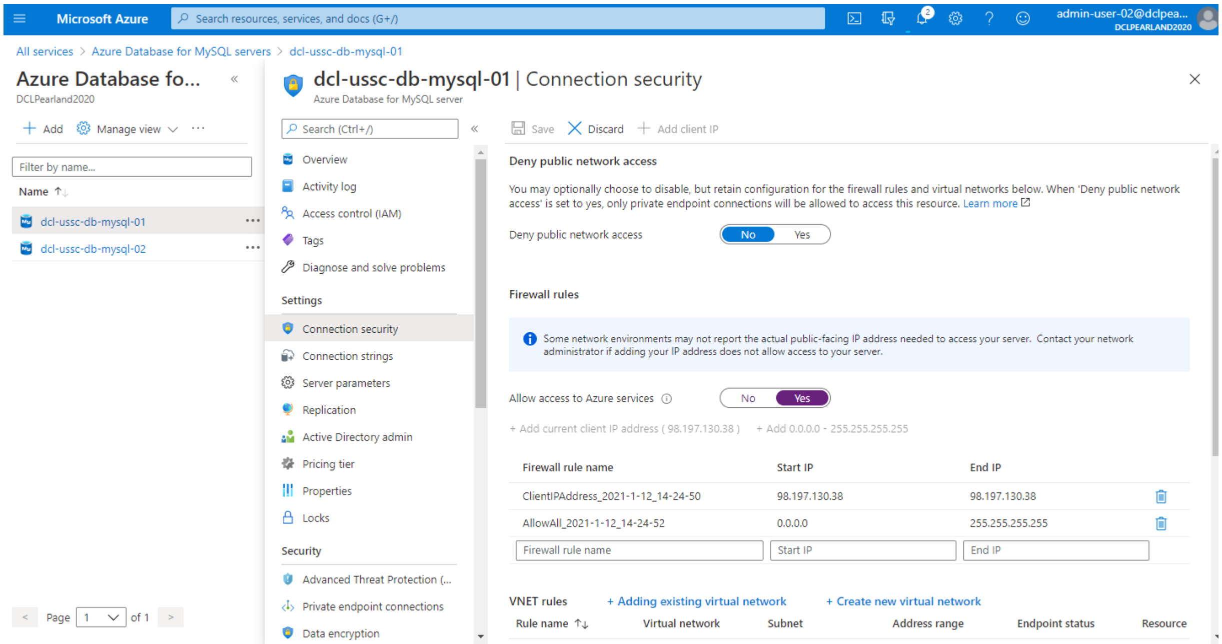
Firewall rule name	Start IP	End IP
Firewall rule name	Start IP	End IP

VNET rules + Adding existing virtual network + Create new virtual network

Rule name	Virtual network	Subnet	Address range	Endpoint status	Resource
No results.					

SSL settings

And then make a change to Firewalls permitting incoming traffic from your IP to the server, like what the screen shot shows below



dcl-ussc-db-mysql-01 | Connection security

Azure Database for MySQL server

Deny public network access

You may optionally choose to disable, but retain configuration for the firewall rules and virtual networks below. When 'Deny public network access' is set to yes, only private endpoint connections will be allowed to access this resource. Learn more

Deny public network access: [✓ No] Yes

Firewall rules

Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server.

Allow access to Azure services: No [✓ Yes]

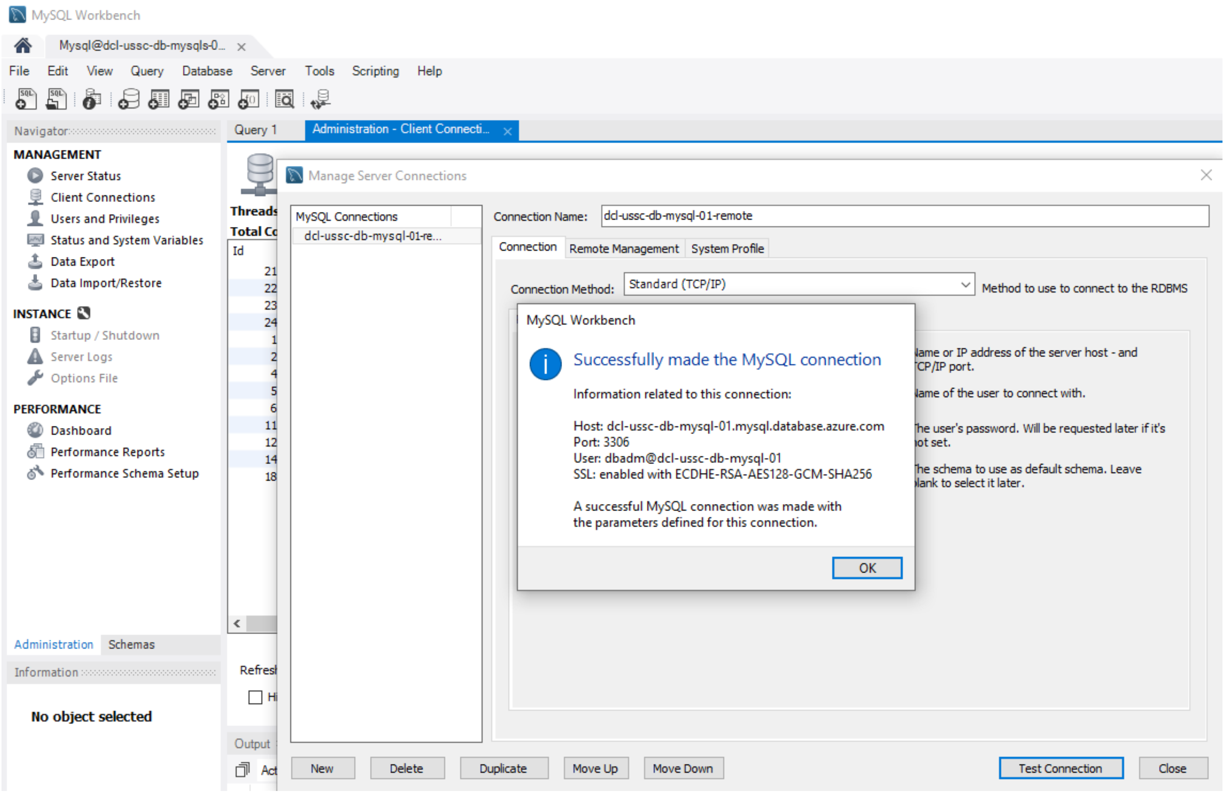
+ Add current client IP address (98.197.130.38)   + Add 0.0.0.0 - 255.255.255.255

Firewall rule name	Start IP	End IP
ClientIPAddress_2021-1-12_14-24-50	98.197.130.38	98.197.130.38
AllowAll_2021-1-12_14-24-52	0.0.0.0	255.255.255.255
Firewall rule name	Start IP	End IP

VNET rules   + Adding existing virtual network   + Create new virtual network

Rule name	Virtual network	Subnet	Address range	Endpoint status	Resource

Do remember save your change by clicking on Save and wait until the update completes. After that, you come back to Workbench to test your connection one more time. This time, your connection will be successful.



Connection Name: dcl-ussc-db-mysql-01-remote

Connection Method: Standard (TCP/IP)   Method to use to connect to the RDBMS

lame or IP address of the server host - and
CP/IP port.

lame of the user to connect with.

he user's password. Will be requested later if it's
ot set.

he schema to use as default schema. Leave
lank to select it later.

MySQL Workbench

Successfully made the MySQL connection

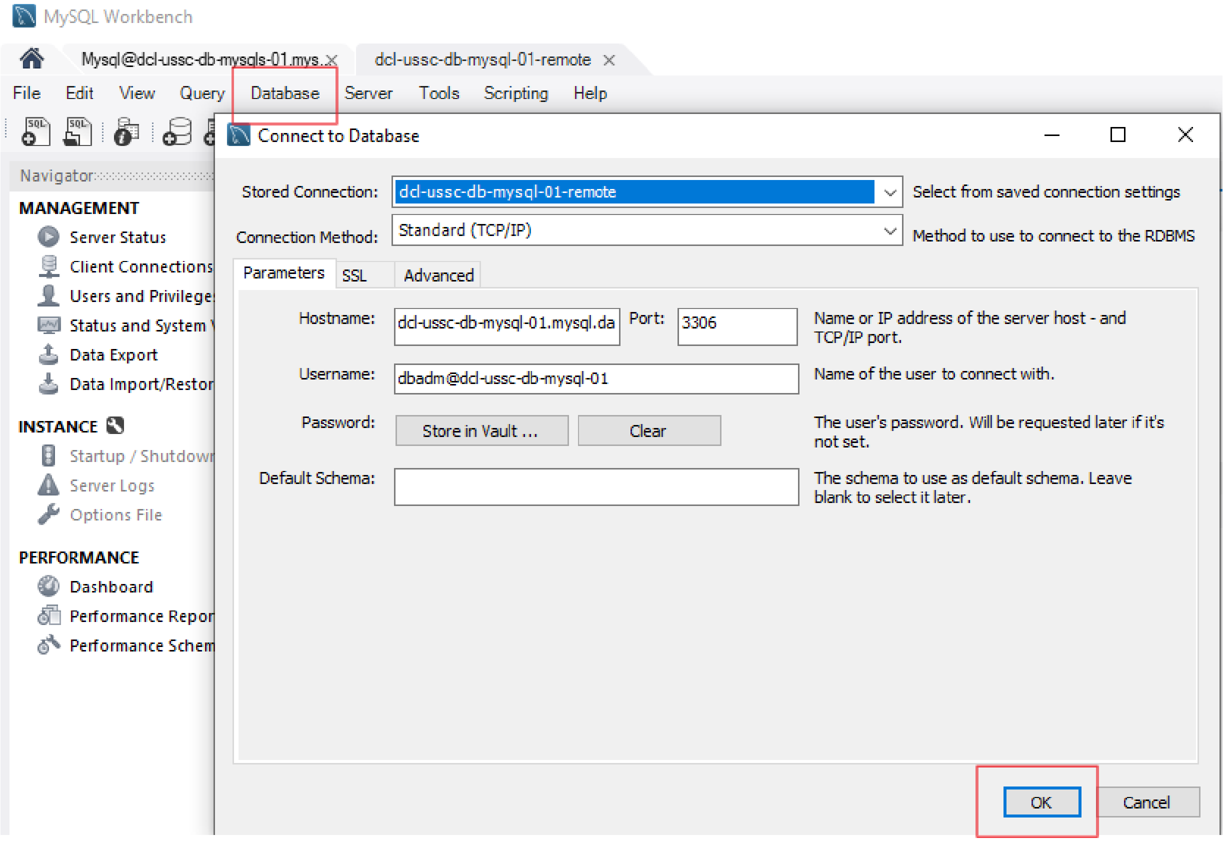
Information related to this connection:

Host: dcl-ussc-db-mysql-01.mysql.database.azure.com
Port: 3306
User: dbadm@dcl-ussc-db-mysql-01
SSL: enabled with ECDHE-RSA-AES128-GCM-SHA256

A successful MySQL connection was made with
the parameters defined for this connection.

OK

While still in Workbench, close the connection test window, click on Databases, select the stored and tested connection that you just tested, and click OK.



Stored Connection: dcl-ussc-db-mysql-01-remote — Select from saved connection settings

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

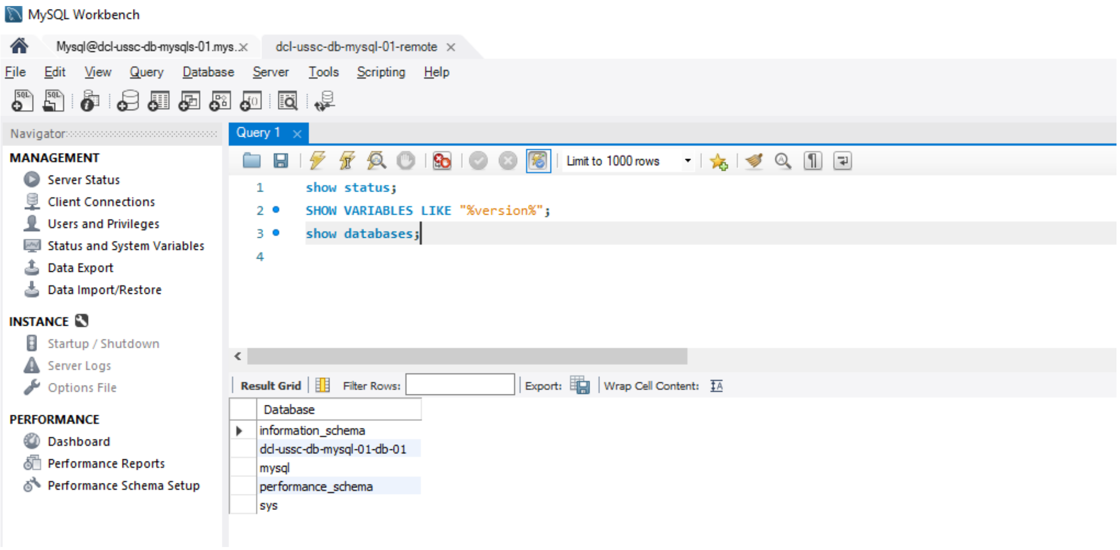
Hostname: dcl-ussc-db-mysql-01.mysql.da Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: dbadm@dcl-ussc-db-mysql-01 — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

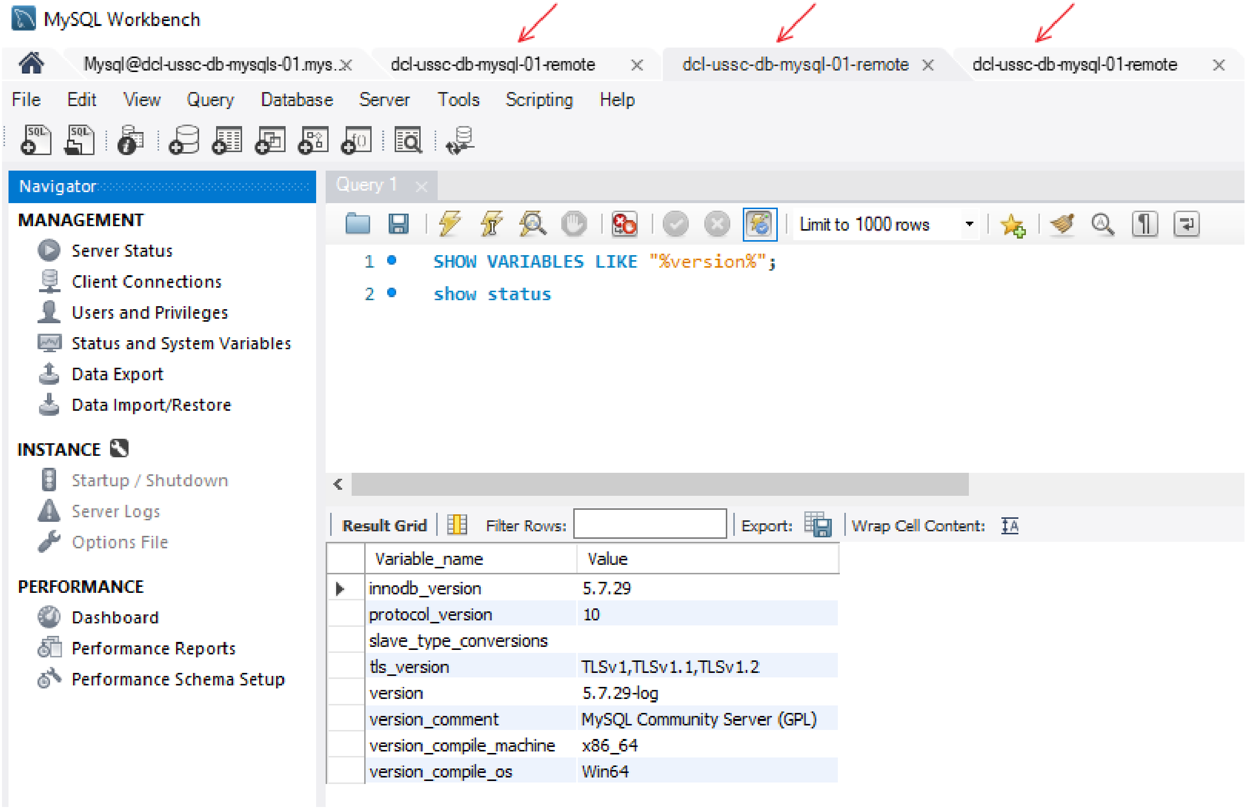
After Workbench connects to the server, it opens up Query editor. In the editor, enter a couple of MySQL commands, and then click the lightning icon or Ctrl-Enter to execute it, it returns query results like this below



```
show status;
SHOW VARIABLES LIKE "%version%";
show databases;
```

Database
information_schema
dcl-ussc-db-mysql-01-db-01
mysql
performance_schema
sys

Workbench allows you to open multiple connections either to one or more than one MySQL servers. Each connection shows up in a tab, you can switch from one connection to another by selecting the desired connection tab. This way you may multi-task working with multiple servers at a time.



```
SHOW VARIABLES LIKE "%version%";
show status
```

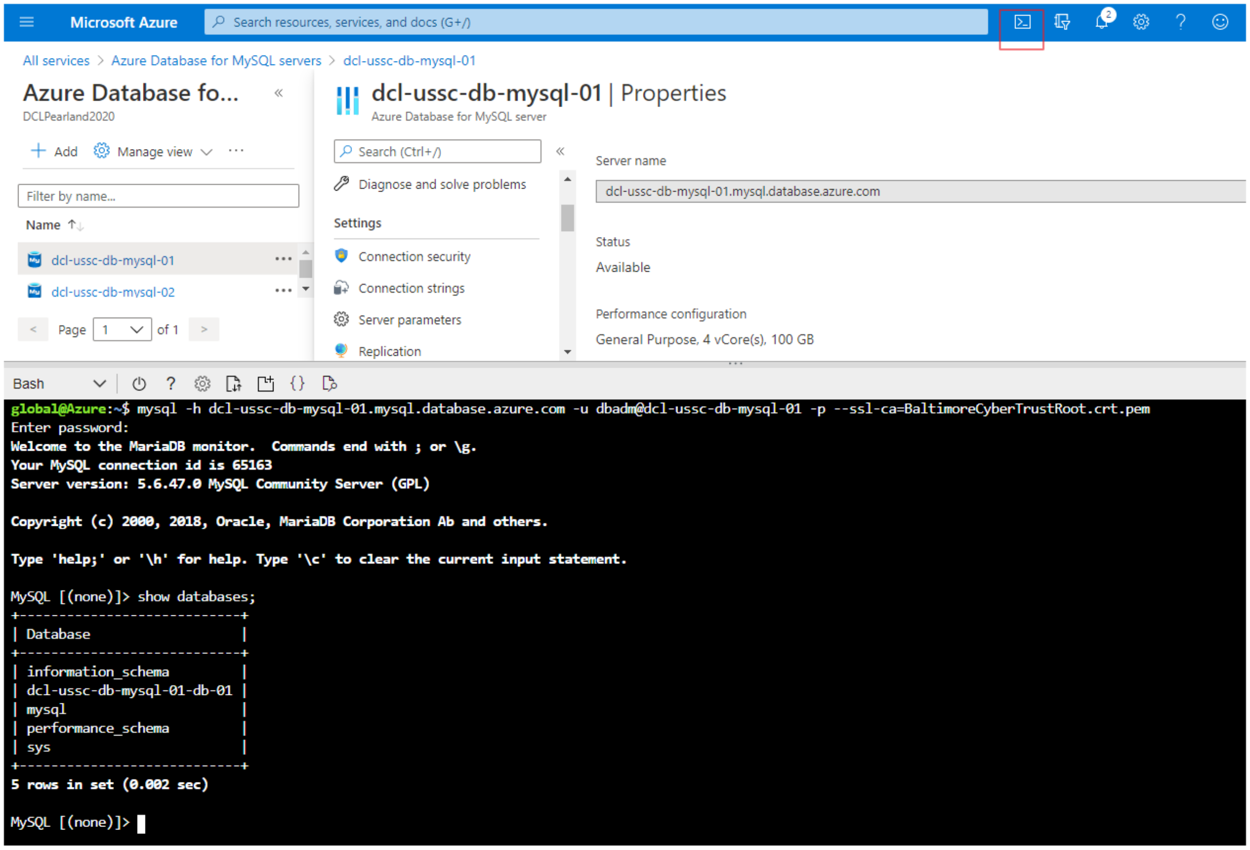
Variable_name	Value
innodb_version	5.7.29
protocol_version	10
slave_type_conversions	
tls_version	TLSv1,TLSv1.1,TLSv1.2
version	5.7.29-log
version_comment	MySQL Community Server (GPL)
version_compile_machine	x86_64
version_compile_os	Win64

Connecting using Azure Portal Cloud Shell

When in Azure Portal, you can open Cloud Shell by clicking its icon in the top bar. See the icon in red square in the screen shot below. Once you have opened Cloud Shell, upload the required SSL PEM file, which was the one you downloaded for Workbench to use previously.

When in Cloud Shell, the command to connect to a server looks like this below
global@Azure:~$ mysql -h dcl-ussc-db-mysql-01.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-01 -p --ssl-ca=**BaltimoreCyberTrustRoot.crt.pem**

The screen shot below shows you successfully logged in the server, and ran 'show databases' command to list all available databases. Do remember ending the command with a semi-colon before pressing Enter key to run it.



Server name
dcl-ussc-db-mysql-01.mysql.database.azure.com

Status
Available

Performance configuration
General Purpose, 4 vCore(s), 100 GB

```
global@Azure:~$ mysql -h dcl-ussc-db-mysql-01.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-01 -p --ssl-ca=BaltimoreCyberTrustRoot.crt.pem
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 65163
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+---------------------------+
| Database                  |
+---------------------------+
| information_schema        |
| dcl-ussc-db-mysql-01-db-01 |
| mysql                     |
| performance_schema        |
| sys                       |
+---------------------------+
5 rows in set (0.002 sec)

MySQL [(none)]>
```

Connecting using MySQL CLI on workstation

To connect to your server in Azure using MySQL CLI that you installed on your local workstation, simply open a Command Prompt window, and type the same command as you used in Cloud Shell, like this below:

```
C:\Users\Tony>mysql -h dcl-ussc-db-mysql-01.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-01 -p --ssl-ca=Downloads\BaltimoreCyberTrustRoot.crt.pem
Enter password: *************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 65195
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> show databases;
+----------------------------+
| Database                   |
+----------------------------+
| information_schema         |
| dcl-ussc-db-mysql-01-db-01 |
| mysql                      |
| performance_schema         |
| sys                        |
+----------------------------+
5 rows in set (0.03 sec)

mysql>
```

To close the session, type 'quit'

```
mysql> quit
Bye

C:\Users\Tony>
```

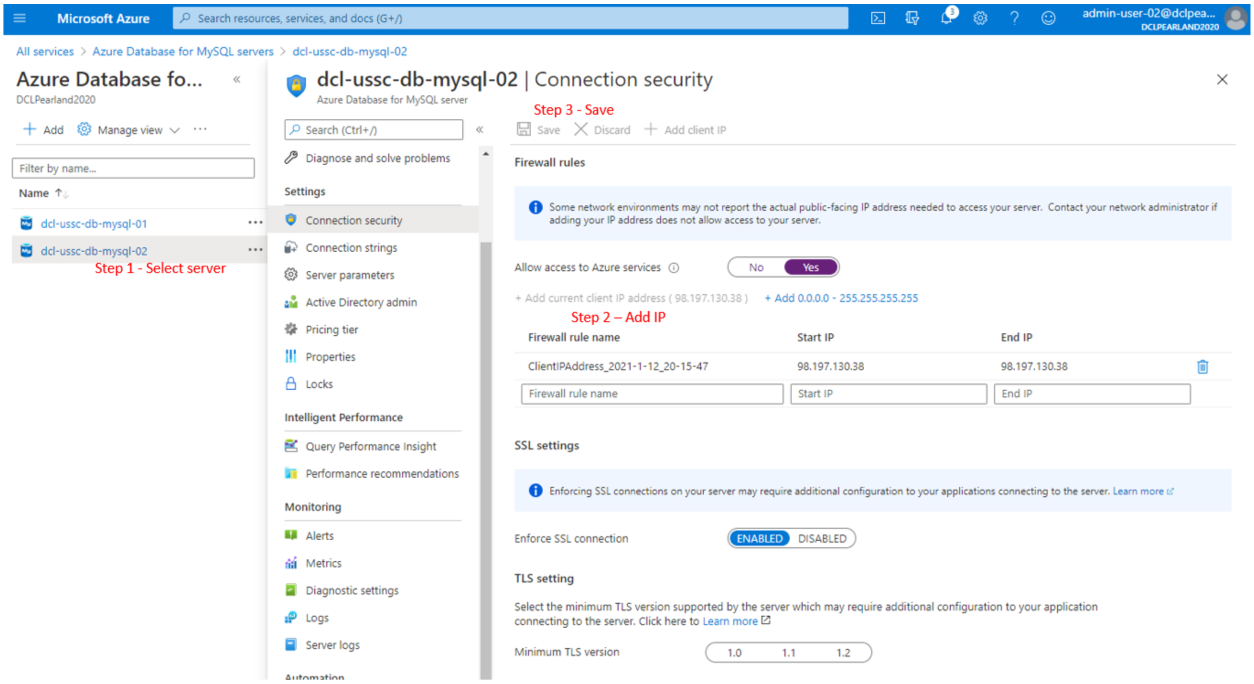
To open another session to connect to the same server or to a different server, open a new Command Prompt window, and run the same command as you used before, replacing server DNS, admin user login name, and password accordingly should they apply. SSL certificate file, however, remains the same with its full path.

This command below was attempted to connect to a different server.

```
C:\Users\Tony>mysql -h dcl-ussc-db-mysql-02.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-02 -p --ssl-ca=Downloads\BaltimoreCyberTrustRoot.crt.pem
Enter password: *************
ERROR 9000 (HY000): Client with IP address '98.197.130.38' is not allowed to connect to this MySQL server.
```

If you see an error like what was shown above, it tells you that server **dcl-ussc-db-mysql-02.mysql.database.azure.com** firewall does not allow incoming traffic from your workstation's IP. To change that, go to Azure Portal, and take three steps.

Step 1 – Select the server
Step 2 – Add the IP in the firewall rule
Step 3 – Save the change



Step 1 - Select server

Step 2 – Add IP

Step 3 - Save

Firewall rules

Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server.

Allow access to Azure services: No [✓ Yes]

Firewall rule name	Start IP	End IP
ClientIPAddress_2021-1-12_20-15-47	98.197.130.38	98.197.130.38
Firewall rule name	Start IP	End IP

SSL settings

Enforcing SSL connections on your server may require additional configuration to your applications connecting to the server. Learn more

Enforce SSL connection: [✓ ENABLED] DISABLED

TLS setting

Select the minimum TLS version supported by the server which may require additional configuration to your application connecting to the server. Click here to Learn more

Minimum TLS version: 1.0 1.1 1.2

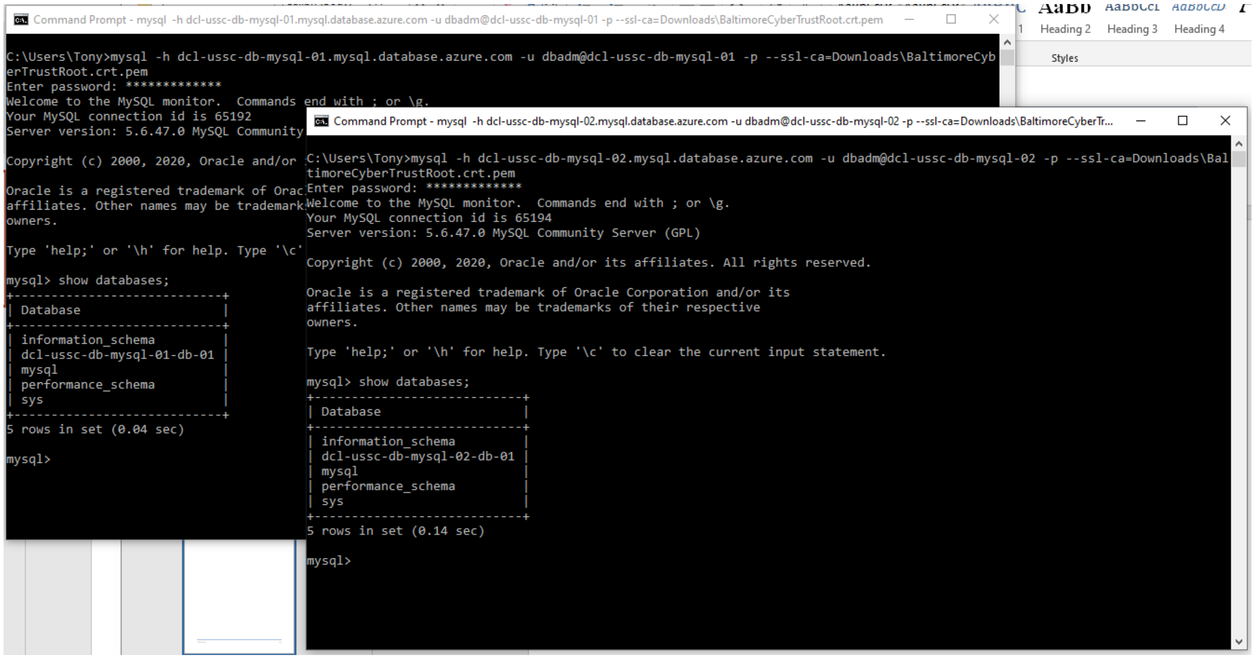
And then come back to mysql and try again.

```
C:\Users\Tony>mysql -h dcl-ussc-db-mysql-02.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-02 -p --ssl-ca=Downloads\BaltimoreCyberTrustRoot.crt.pem
Enter password: ************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 65194
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+---------------------------+
| Database                  |
+---------------------------+
| information_schema        |
| dcl-ussc-db-mysql-02-db-01 |
| mysql                     |
| performance_schema        |
| sys                       |
+---------------------------+
```

5 rows in set (0.14 sec)

mysql>

Now you have two sessions open, connecting to two different servers simultaneously.


Command Prompt - mysql -h dcl-ussc-db-mysql-01.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-01 -p --ssl-ca=Downloads\BaltimoreCyberTrustRoot.crt.pem
Command Prompt - mysql -h dcl-ussc-db-mysql-02.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-02 -p --ssl-ca=Downloads\BaltimoreCyberTr...
AaBb
AaBbCcI
AaBbCcD
Heading 2
Heading 3
Heading 4
Styles

```
C:\Users\Tony>mysql -h dcl-ussc-db-mysql-01.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-01 -p --ssl-ca=Downloads\BaltimoreCyb
erTrustRoot.crt.pem
Enter password: *************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 65192
Server version: 5.6.47.0 MySQL Community

Copyright (c) 2000, 2020, Oracle and/or

Oracle is a registered trademark of Orac
affiliates. Other names may be trademark
owners.

Type 'help;' or '\h' for help. Type '\c'

mysql> show databases;
+---------------------------+
| Database                  |
+---------------------------+
| information_schema        |
| dcl-ussc-db-mysql-01-db-01 |
| mysql                     |
| performance_schema        |
| sys                       |
+---------------------------+
5 rows in set (0.04 sec)

mysql>
```

```
C:\Users\Tony>mysql -h dcl-ussc-db-mysql-02.mysql.database.azure.com -u dbadm@dcl-ussc-db-mysql-02 -p --ssl-ca=Downloads\Bal
timoreCyberTrustRoot.crt.pem
Enter password: *************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 65194
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+---------------------------+
| Database                  |
+---------------------------+
| information_schema        |
| dcl-ussc-db-mysql-02-db-01 |
| mysql                     |
| performance_schema        |
| sys                       |
+---------------------------+
5 rows in set (0.14 sec)

mysql>
```

Cleaning up resources

You may avoid unnecessary service usage and billing charges by cleaning up all the resources that you created and used in a walkthrough like the example case we did so far. The quickest way to do the cleaning up is to remove RGs that you used for your MySQL. As an example, simply type the following two commands in Azure CLI, substituting the RG names with yours should they be different.

```
C:\Users\Tony>az group delete -n dcl-ussc-rg-01 --yes --no-wait

C:\Users\Tony>az group delete -n dcl-ussc-rg-02 --yes --no-wait

C:\Users\Tony>
```

Summary

In this write-up, we talked you through about how to set up MySQL server instances in Azure using both Azure Portal and Azure CLI and how to use MySQL client on your local workstation to connect to your MySQL server(s) in Azure. We also discussed MySQL client and its usage in detail. MySQL client comes in two forms. One is Workbench in GUI, the other, mysql in command line interface (CLI). You can make multiple connections to multiple servers simultaneously using either Workbench or CLI. For security, SSL-enabled is the server's default. SSL certificate is required to connect to an SSL-enabled server in

Azure. SSL certificate can be readily obtained from Azure. MySQL client allows you to work with multiple MySQL servers in the same time. Using MySQL client, you may create database(s) in each server. For how to use MySQL database to store, update, and query your data, and how to connect to MySQL server and database from within your applications, please refer to Azure Database for MySQL documentation that Microsoft provides and MySQL documentation that Oracle provides.

References

Azure Database for MySQL Documentation