

Hands-On Objective: Azure SQL Server and Azure SQL Database

Objective

The objective of this hands-on exercise is to create an Azure SQL Server and an Azure SQL Database, configure firewall rules to allow access, connect to the database using Azure Query Editor and a client tool, execute SQL queries, and finally terminate all created resources.

Goals of the Hands-On

1. Create an Azure resource group
 2. Create an Azure SQL Server
 3. Create an Azure SQL Database
 4. Configure firewall rules for SQL server access
 5. Connect to the database using Azure Query Editor
 6. Connect to the database using a client tool
 7. Terminate all Azure resources
-

A) Creation of Resource Group

The Azure portal was accessed using a valid Azure account.

From the Azure search bar, **Resource Groups** was selected and a new resource group was created with the following details:

- **Subscription:** Free Trial / Pay-As-You-Go
- **Resource Group Name:** GL-SQL-RG
- **Region:** Default

The resource group was reviewed and created successfully.

B) Creation of Azure SQL Server

An Azure SQL Server was created using the following configuration:

- **Subscription:** Free Trial / Pay-As-You-Go
- **Resource Group:** GL-SQL-RG
- **Server Name:** glsqlserver01
- **Location:** Default
- **Authentication Method:** SQL authentication
- **Server Admin Login:** mbadmin
- **Password:** User-defined secure password

The configuration was reviewed and the SQL Server was created successfully.

C) Creation of Azure SQL Database

An Azure SQL Database was created with the following details:

- **Subscription:** Free Trial / Pay-As-You-Go
- **Resource Group:** GL-SQL-RG
- **Database Name:** SampleDB
- **Server:** glsqlserver01
- **Data Source:** Use existing data
- **Sample Data:** Sample

The database deployment completed successfully.

D) Firewall Configuration for SQL Server Access

To allow access to the SQL Server:

1. Navigated to **SQL Servers**
2. Selected **glsqlserver01**
3. Opened **Firewalls and virtual networks**
4. Clicked **Add Client IP**

5. Saved the configuration

This allowed the client system to access the SQL Server.

E) Connecting to Database via Azure Query Editor

The database was accessed using Azure Query Editor:

1. Navigated to **SQL Databases**
2. Selected **SampleDB**
3. Opened **Query Editor (Preview)**
4. Logged in using SQL authentication

SQL Query Executed

```
SELECT * FROM [SalesLT].[Customer];
```

The query executed successfully and returned customer records on the screen.

F) Connecting to Database via Client Tool

1. Creation of Virtual Machine

A Windows Virtual Machine was created to act as a client system:

- **Subscription:** Free Trial / Pay-As-You-Go
- **Resource Group:** GL-SQL-RG
- **Virtual Machine Name:** GL-SQL-VM
- **Image:** Windows 10 Pro
- **Username:** gladmin
- **Password:** User-defined secure password

The VM was deployed successfully.

2. Firewall Rule for VM Access

To allow the VM to connect to the SQL Server:

- A new firewall rule was added on **glsqlserver01**
- **Rule Name:** AzureVMAccess
- **Start IP:** VM Public IP
- **End IP:** VM Public IP

The rule was saved successfully.

3. Client Tool Installation and Connection

1. Connected to the VM using Remote Desktop
2. Installed **Azure Data Studio**
3. Opened Azure Data Studio
4. Created a new connection with the following details:
 - **Connection Type:** Microsoft SQL Server
 - **Server Name:** Copied from Azure portal
 - **Authentication:** SQL Login
 - **Username:** mbadmin
 - **Password:** Database password

The connection was established successfully.

4. SQL Query Execution via Client Tool

After connecting to the database, a new query window was opened.

SQL Query Executed

```
SELECT * FROM [SalesLT].[Customer];
```

The query executed successfully and displayed the result set.

G) Termination of Resources

To avoid unnecessary costs, all resources were deleted:

1. Navigated to **Resource Groups**
2. Selected **GL-SQL-RG**
3. Clicked **Delete Resource Group**
4. Entered **GL-SQL-RG** to confirm
5. Deleted the resource group

All associated resources were removed successfully.

Result

An Azure SQL Server and Azure SQL Database were successfully created.
The database was accessed using both Azure Query Editor and a client tool, and SQL queries were executed successfully.
All resources were terminated after completion.

Conclusion

This hands-on exercise demonstrated the end-to-end process of deploying, securing, and accessing an Azure SQL Database using multiple connection methods while following proper resource management practices.