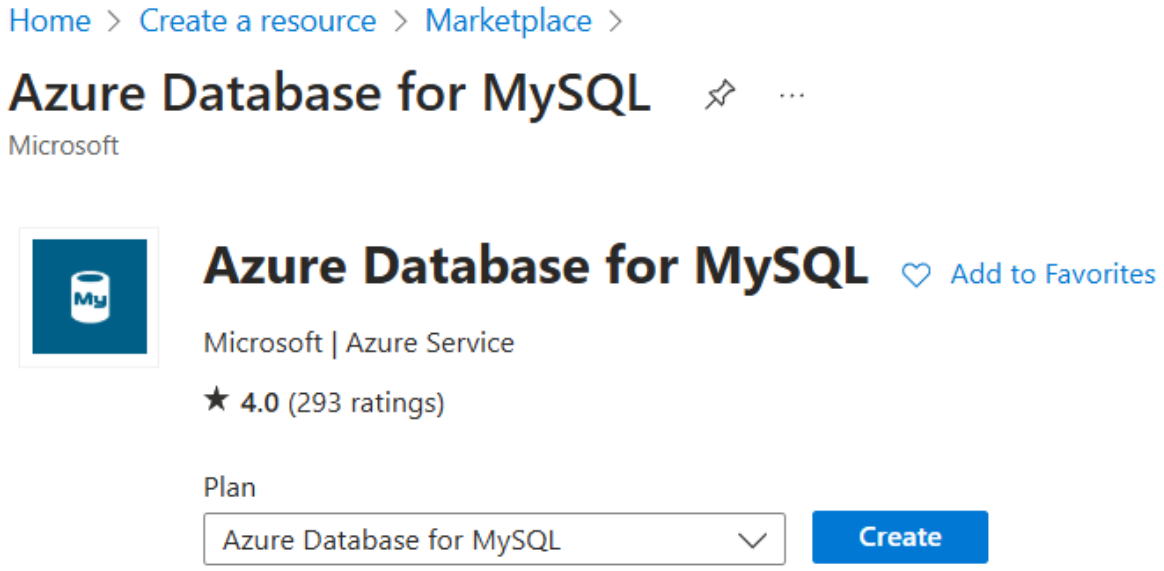
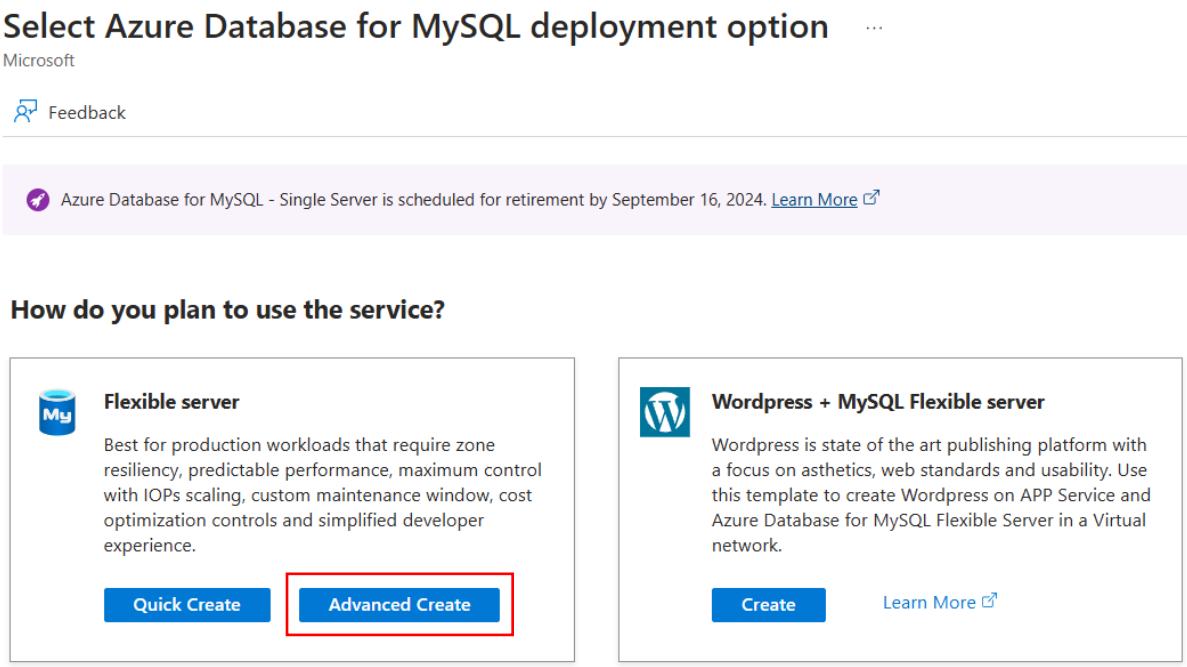
😊 Application to Generate Data

The process involves setting up an Azure Database for MySQL using a Flexible Server in the Azure Portal. After configuring authentication and firewall settings, the server is connected via Azure Data Studio with the MySQL extension. A database (demodb) and a table (logdata) are created. Next, an external program from GitHub is configured with the log file path and server name, then executed to insert data into the table. The end goal is to automate log data entry into MySQL, ensuring efficient storage and retrieval of operational records for further analysis.

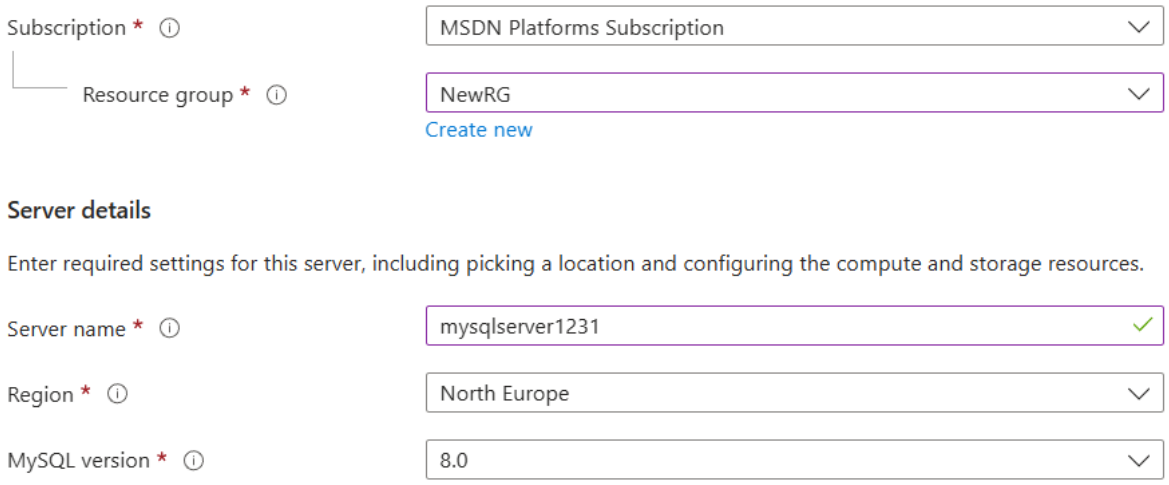
1. In this lab, we are going to Generate Data using the external Application inside the MySQL Database.
2. So, first we must move to the Azure Portal marketplace and search for MySQL and we will choose Azure Database for MySQL.



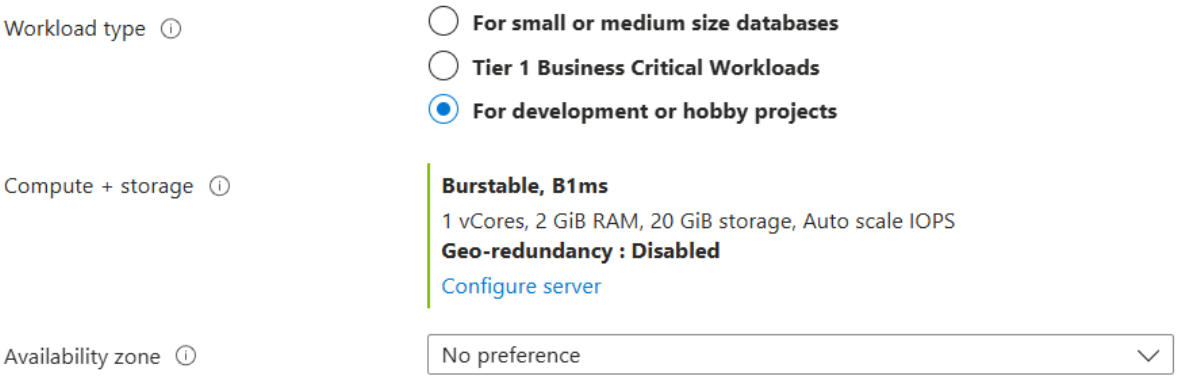
3. We also need to choose our Database from the deployment options. So, we will choose a Flexible server. Click on advanced create.

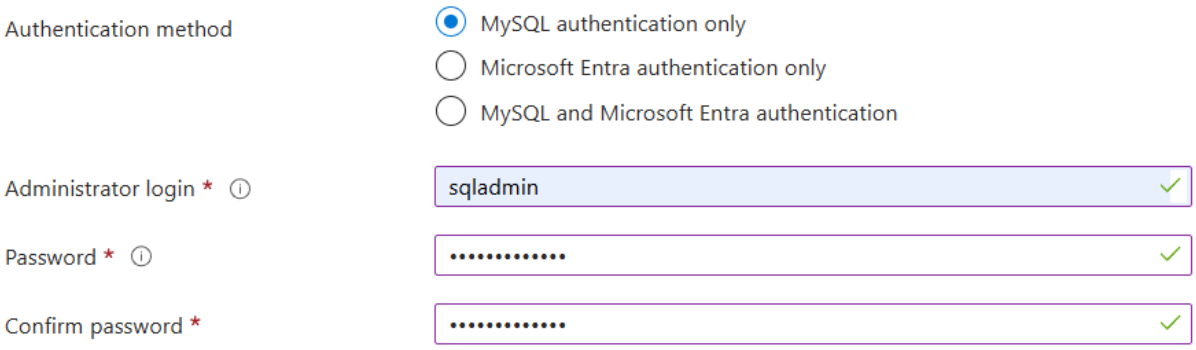


4. Now we need to choose the resource group, give a name to our server then choose the region with the SQL version.

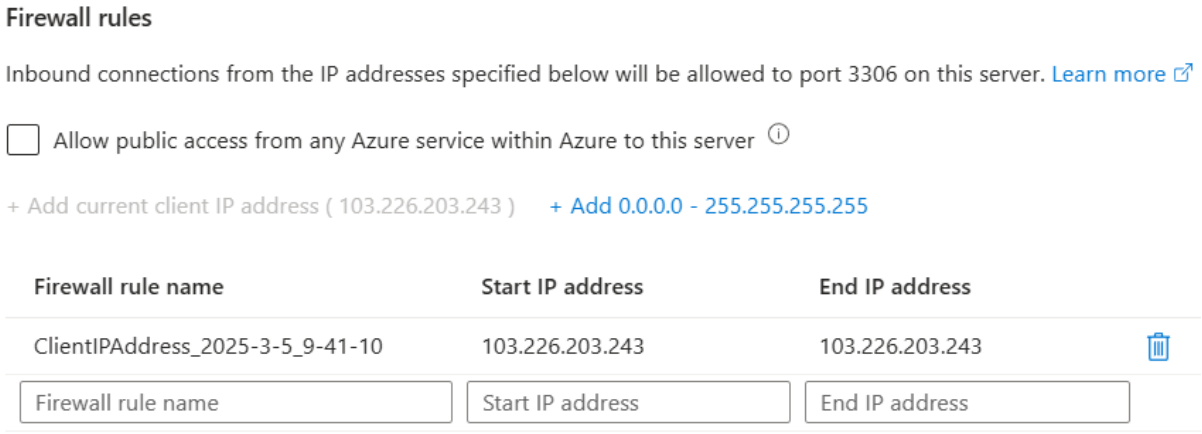


5. We will leave workload type and compute storage to default and choose the authentication as SQL authentication. Then give a username and password.

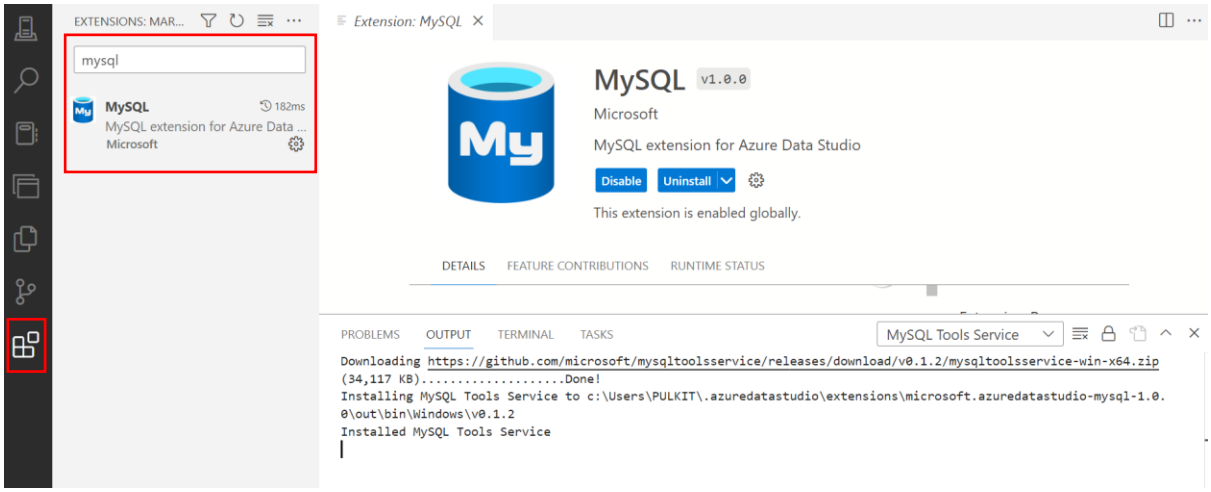




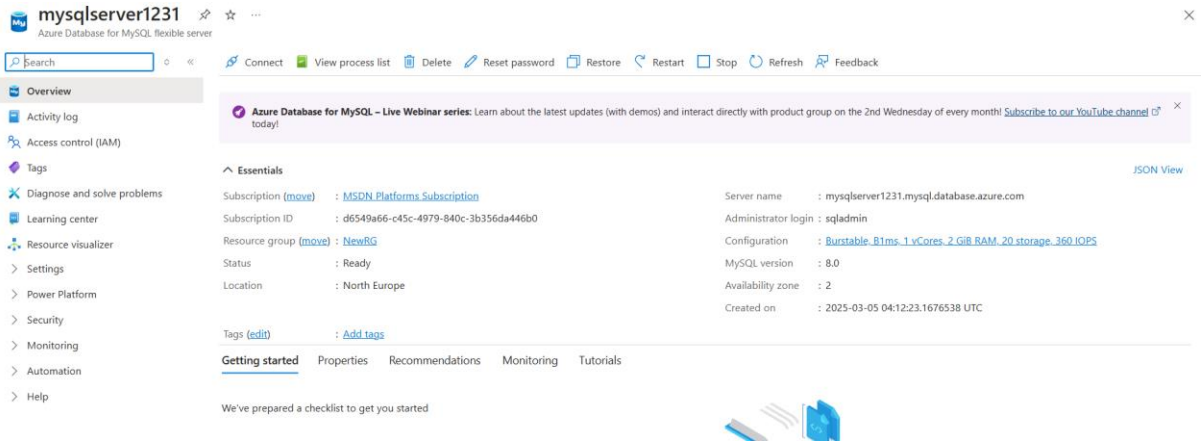
6. In the networking tab just remember to add your IP address in the firewall rules.



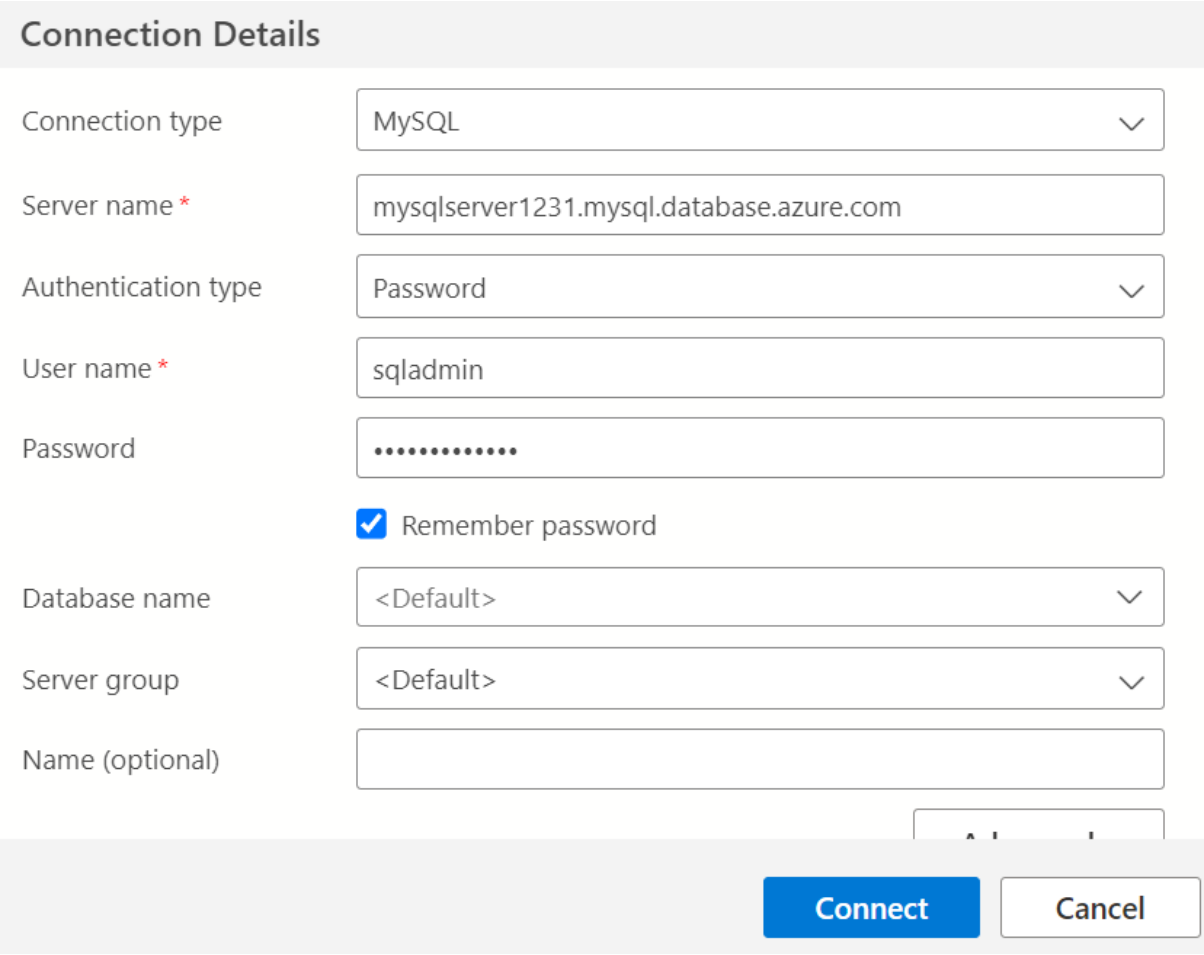
7. Once our server is up and running then we will connect our server using Azure Data Studio. So, open the Azure Data Studio.
8. Then go to the extensions and search for MySQL then install it.



9. Now take the server's name of MySQL database and open Azure Data Studio.



10. Here you need to choose MySQL in the connection type then paste the server's name, after that give the username and password then click on connect.

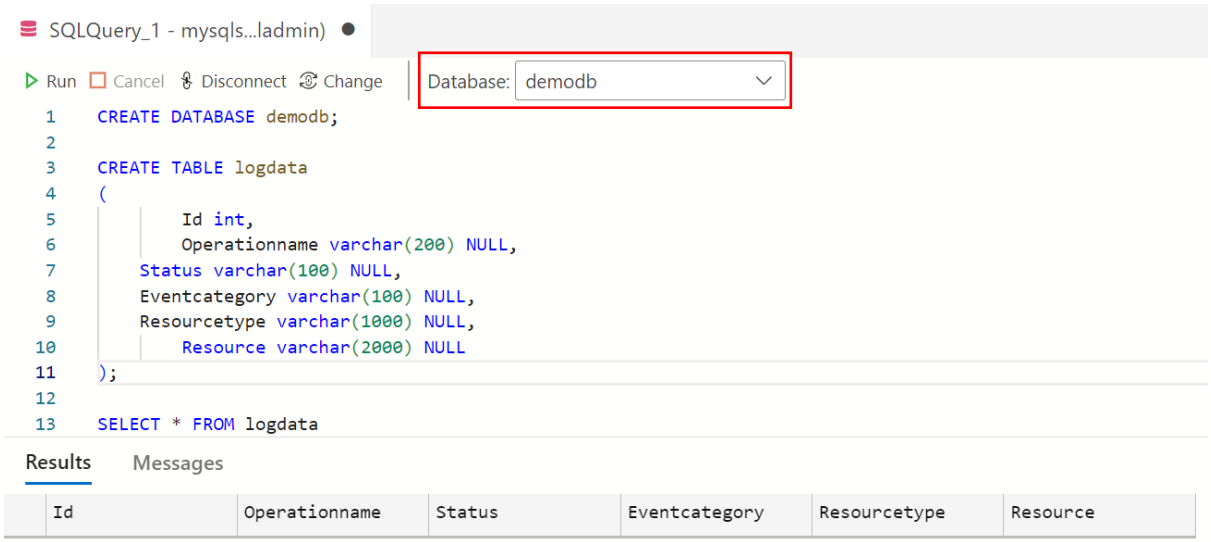


11. Once the server is connected then we need to create a database in it and inside the database we will create a table.
12. Below you can see that first we created our database then we used it. After that we created an empty table.

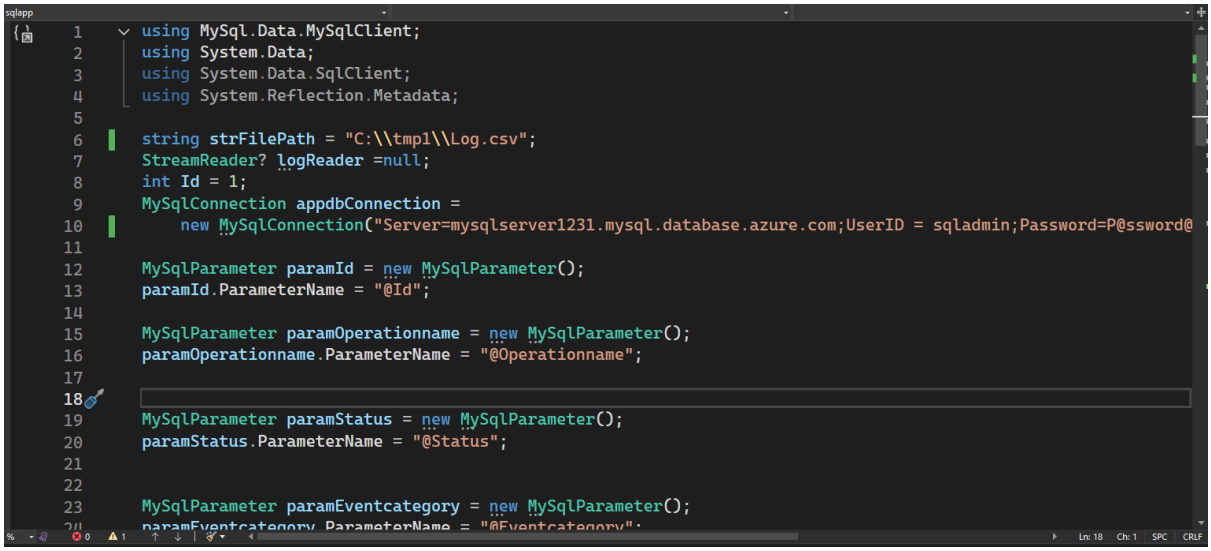
CREATE DATABASE appdb;

USE demodb;

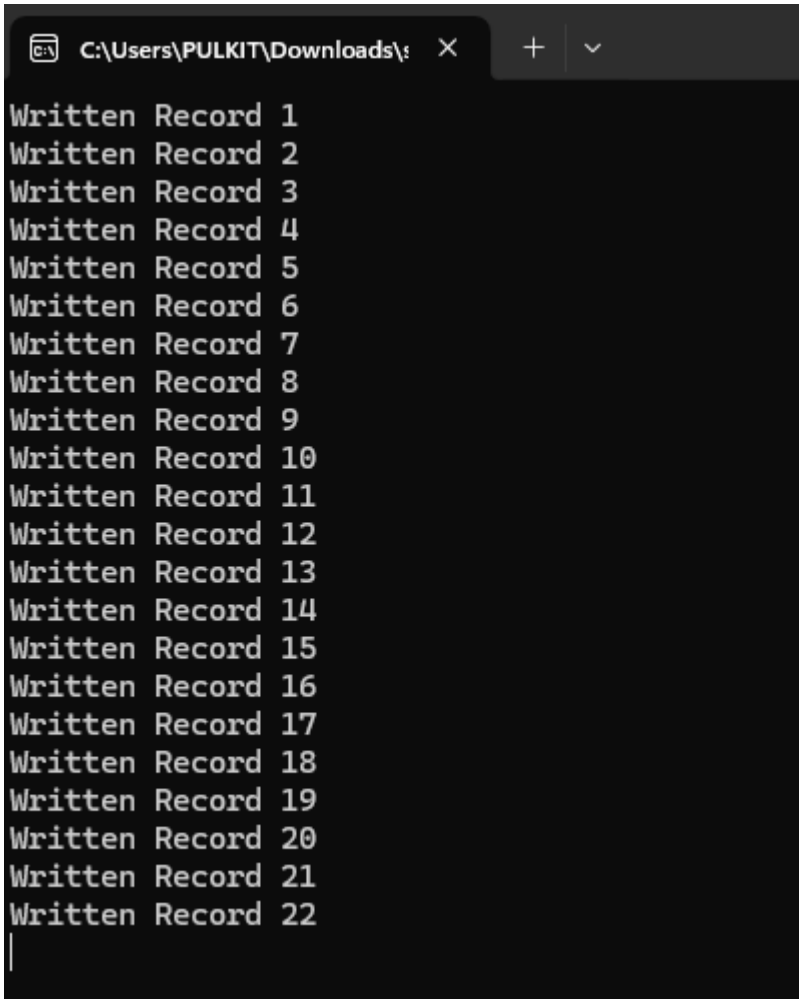
```
CREATE TABLE logdata
(
    Id int,
    Operationname varchar(200) NULL,
    Status varchar(100) NULL,
    Eventcategory varchar(100) NULL,
    Resourcetype varchar(1000) NULL,
    Resource varchar(2000) NULL
);
```



13. Then open up the program you downloaded from GitHub and on line number 6 give the path of the log file you get with the program then on line number 10 you need to give the server name, save the changes and execute your program.



14. If you have done everything properly then you will see that records are being written onto the table and if you try to view the data inside the table then you will have information accordingly.



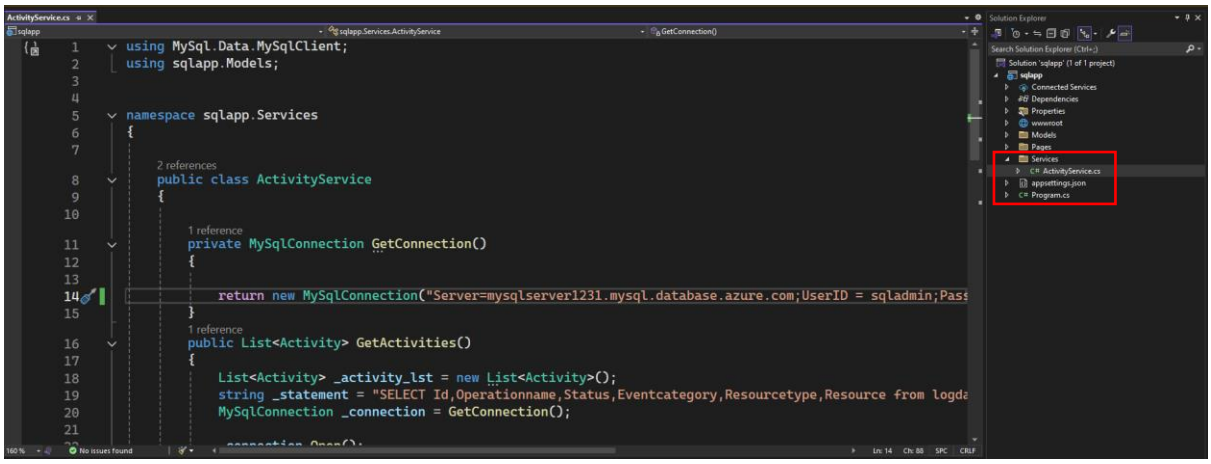
Run Cancel Disconnect Change Database: demodb

```
1   CREATE DATABASE demodb;
2
3   CREATE TABLE logdata
4   (
5           Id int,
6           Operationname varchar(200) NULL,
7       Status varchar(100) NULL,
8       Eventcategory varchar(100) NULL,
9       Resourcetype varchar(1000) NULL,
10          Resource varchar(2000) NULL
11  );
12
13  SELECT * FROM logdata
```

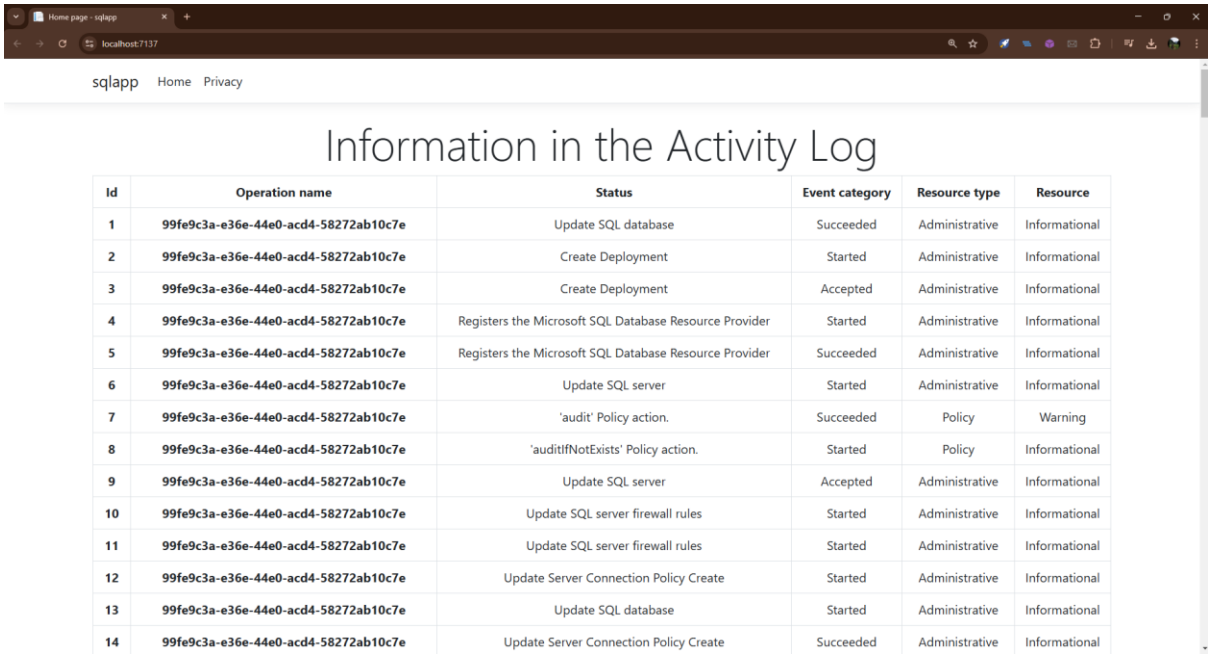
Results Messages

	Id	Operationname	Status	Eventca
1	1	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Succ
2	2	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Start
3	3	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Accep
4	4	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Start
5	5	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Succ
6	6	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Start
7	7	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'audit' Policy action.	Succ
8	8	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'auditIfNotExists' Policy action.	Start

15. Then on GitHub you can find one more program and if you open up this program in Visual Studio change the server's name in Activity Services save it and then run the application.



16. You will see that this application is helping you to display the application onto the browser from your MySQL database.



Id	Operation name	Status	Event category	Resource type	Resource
1	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Succeeded	Administrative	Informational
2	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Started	Administrative	Informational
3	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Accepted	Administrative	Informational
4	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Started	Administrative	Informational
5	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Succeeded	Administrative	Informational
6	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Started	Administrative	Informational
7	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'audit' Policy action.	Succeeded	Policy	Warning
8	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'auditIfNotExists' Policy action.	Started	Policy	Informational
9	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Accepted	Administrative	Informational
10	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server firewall rules	Started	Administrative	Informational
11	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server firewall rules	Started	Administrative	Informational
12	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update Server Connection Policy Create	Started	Administrative	Informational
13	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Started	Administrative	Informational
14	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update Server Connection Policy Create	Succeeded	Administrative	Informational