

# Azure SQL Database

## Objectives

In this hands-on lab, you will learn how to:

- Create an Azure SQL Database Server
- Create a SQL Database
- Configure the firewall
- Add a user to the SQL Server and Database
- Modify your sample web application to enable user registration and login

## Prerequisites

The following are required to complete this hands-on lab:

- An active Microsoft Azure subscription
- Visual Studio 2017 Community or greater (Professional or Enterprise)

## Exercises

This hands-on lab includes the following exercises:

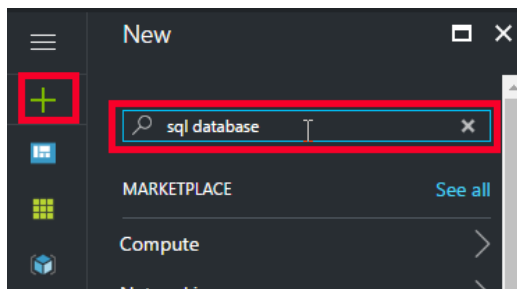
- Exercise 1: Create an Azure SQL database
- Exercise 2: Configure the firewall
- Exercise 3: Add a user to the database
- Exercise 4: Configure Web Application to the use Azure SQL database

Estimated time to complete this lab: 15 - 25 minutes

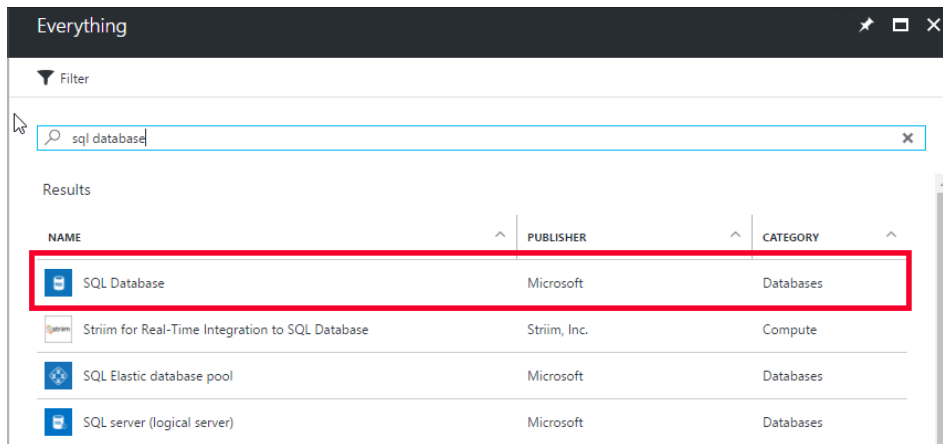
## Exercise 1: Create an Azure SQL Database

In this exercise, you will create an Azure SQL server and database to use later with the sample web application.

1. Go the Azure Portal <https://portal.azure.com> and sign into your Azure account
2. Click the + New (or use the keyboard shortcut N) to open the **New blade**, then type **sql database** and hit **enter**

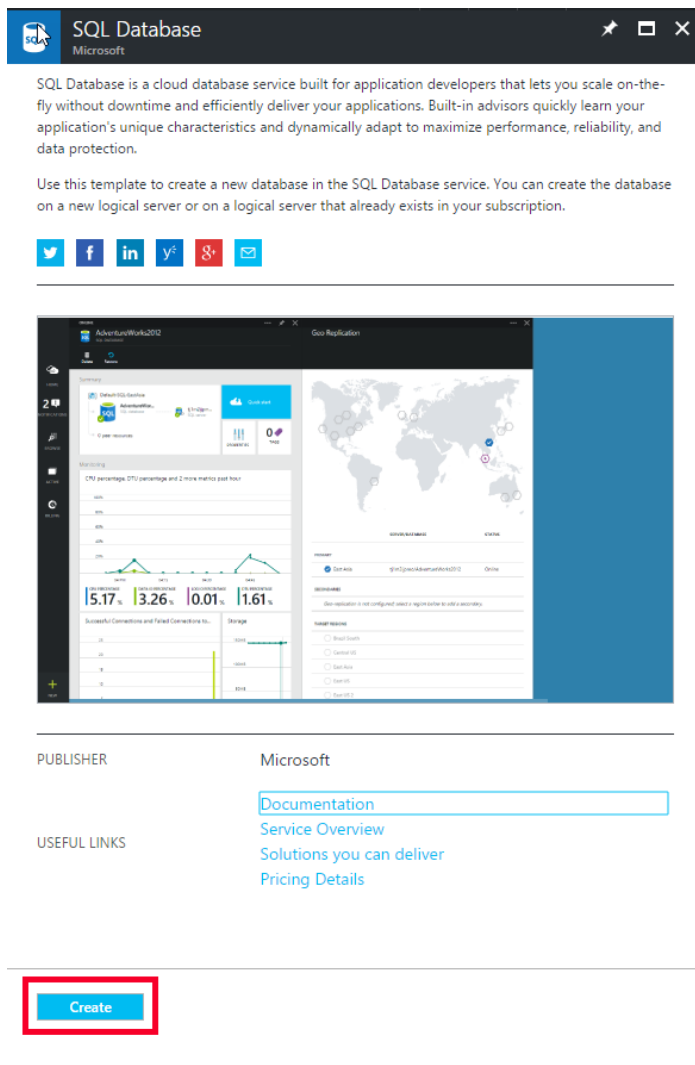


This will open the search results blade.



### 3. Select **SQL Database**

This will show the information about the SQL database product.



### 4. Click **Create**

This will open the SQL Database blade.

5. On the SQL Database blade, fill out the following selections:
  - **Database name:** GABDB<first name>
  - **Subscription:** choose your subscription
  - **Resource Group:** Create new
    - Enter name of: GABDataLab
  - **Select source:** Blank database
6. Click on the **Server** item, to create a new server
7. In the New Server blade, enter the following information:
  - **Server name:** gabdbserver<first name>
  - **Server admin login:** <your first name>
  - **Password:** <a strong password minimum of 12 characters>
  - **Confirm Password:** retype your password
  - **Location:** East US
  - **Allow azure services to access server:** leave checked

New server

\* Server name  
gabdbserverjason ✓  
.database.windows.net

\* Server admin login  
jason ✓

\* Password  
..... ✓

\* Confirm password  
..... ✓

\* Location  
East US ▼

☒ Allow azure services to access server ⓘ

Select

8. Click the **Select** button

9. Back on the SQL Database blade, select the **Pricing tier** item.

This opens the (somewhat new) Configure performance blade

Configure performance

Feedback

Basic

For infrequent access and less demanding workloads

5 DTU

Starting at 4.99 USD / month

Standard

For most production workloads

10-100 DTU

Starting at 15.00 USD / month

Premium

For IO-intensive workloads and highest availability

125-4000 DTU

Starting at 465.00 USD / month

PremiumRS

For IO-intensive workloads but with a limited availability guarantee (Preview)

125-1000 DTU

Starting at 116.25 USD / month

DTU (10-100 DTU) - [What is a DTU?](#)

20 (S1)

Monthly

30.00 USD

Storage (100 MB-250 GB)

250 GB

Included

0.00 USD

Monthly cost

30.00 USD

Apply

10. Change the **DTU setting to 20** by dragging the handle to the left
11. Click on the **Apply** button
12. Back on the SQL Database blade, **check the Pin to dashboard** and click the **Create** button to start the creation of the server and blank database.

SQL Database

\* Database name

GABDBJason

✓

\* Subscription

Visual Studio Enterprise - MVP License

▼

\* Resource group ⓘ

☒ Create new ☐ Use existing

GABDataLab

✓

\* Select source ⓘ

Blank database

▼

\* Server

gabdbserverjason (East US)

>

Want to use SQL elastic pool? ⓘ

☐ Yes ☒ Not now

\* Pricing tier ⓘ

Standard S1: 20 DTU, 250 GB

>

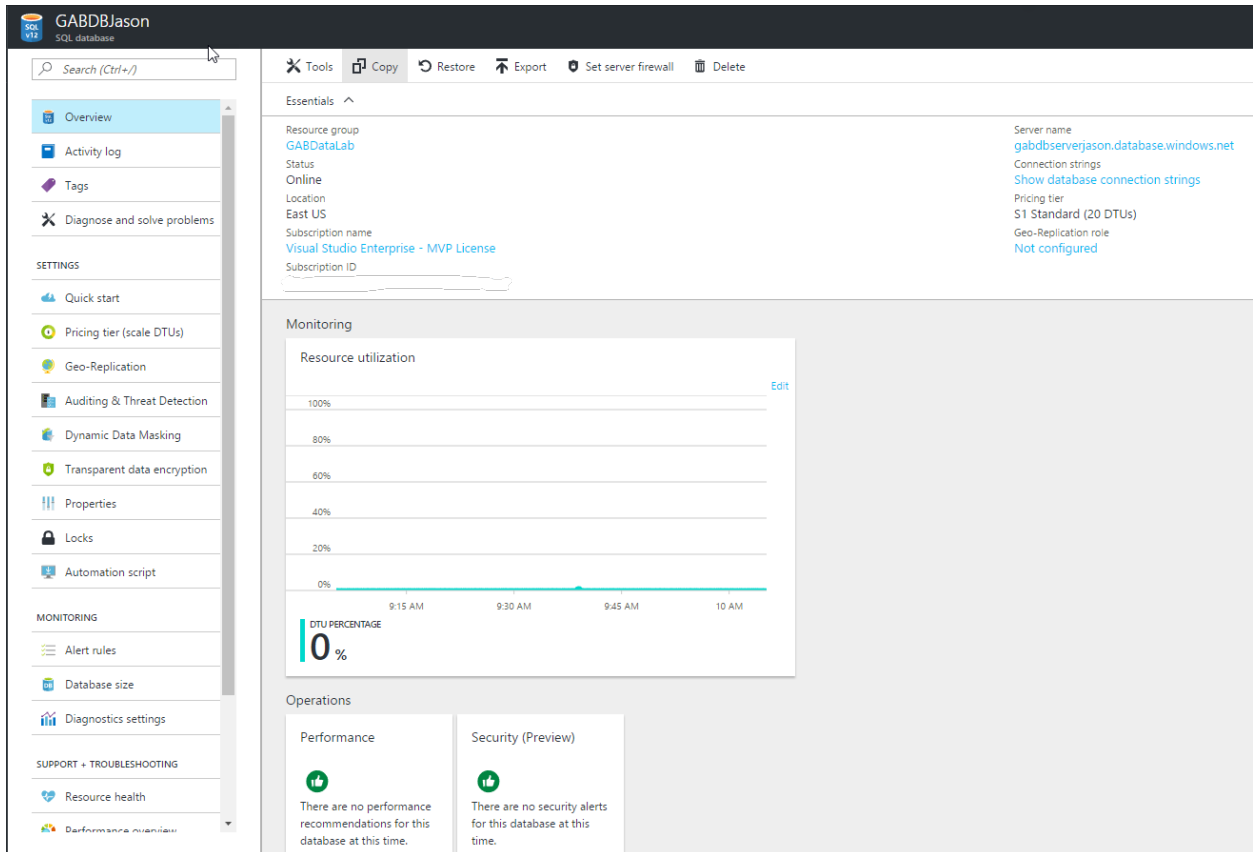
\* Collation ⓘ

SQL\_Latin1\_General\_CP1\_CI\_AS

☒ Pin to dashboard

CreateAutomation options

Once the database gets created (should take 1-3 minutes), the database blade should open to the overview.

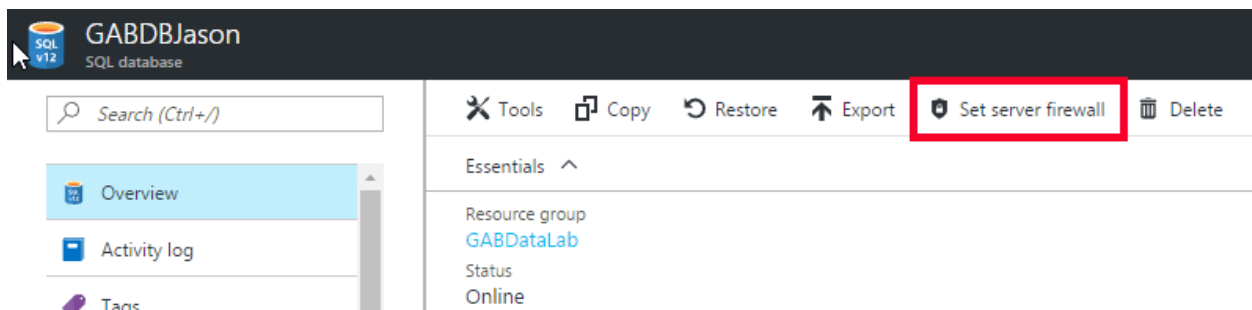


The next step is to configure the firewall so you can access the database from your local computer.

## Exercise 2: Configure the firewall

In this exercise, you will configure your Azure SQL firewall to allow access from your local machine.

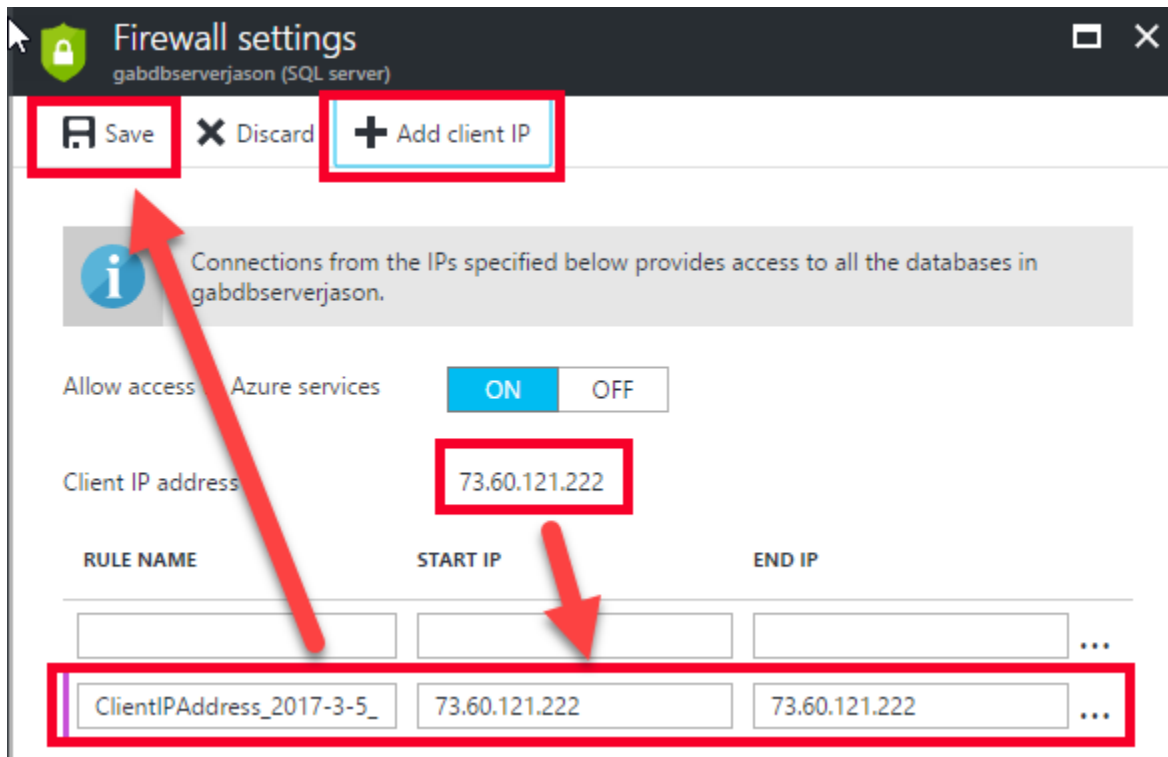
1. In the Azure Portal, go to your SQL Database overview pane (if it isn't already showing)
2. In the middle of the overview tool bar, select the **Set server firewall** button



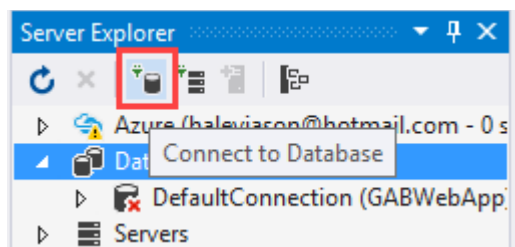
This will show the Firewall settings blade.

3. In the blade's toolbar, click on the **+ Add client IP** button.

This will take your client IP address and add a rule for you – or you can type the ip address yourself. Like almost all resources in Azure, the firewall rule has a name.

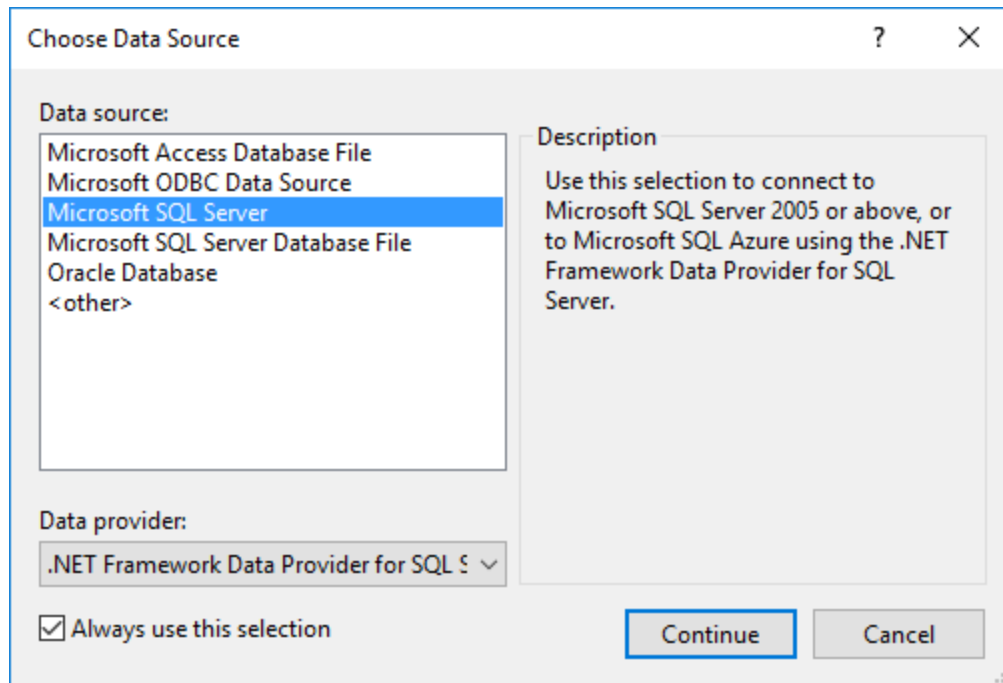


4. Click the **Save** button and you should get a confirmation dialog indicating the rule was added.
5. Close the Firewall settings blade
6. Back in your **Visual Studio 2017**, Open your **Server Explorer** (View -> Server Explorer)
7. In the toolbar, click on the **Connect to Database** button



This will show the Choose Data Source dialog box





8. Select **Microsoft SQL Server** and click **Continue**
9. Enter the following information into the **Add Connection** dialog
  - **Server Name:** gabdbserver<firstname>.database.windows.net
  - **Authentication:** SQL Server Authentication
  - **User name:** <sql server admin you configured in Exercise 1, step 7>
  - **Password:** <password for the sql server admin>
  - **Select or enter a database name:** master

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient) Change...

Server name: gabdbserverjason.database.windows.net Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: jason

Password: ..... Save my password ☒

Connect to a database

☒ Select or enter a database name: master

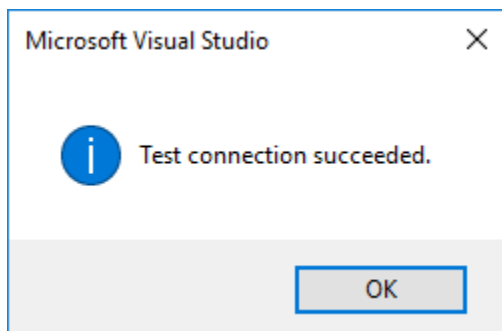
☐ Attach a database file: Browse...

Logical name:

Advanced...

Test Connection OK Cancel

10. Click the **Test Connection** button to test if the firewall will allow you to connect



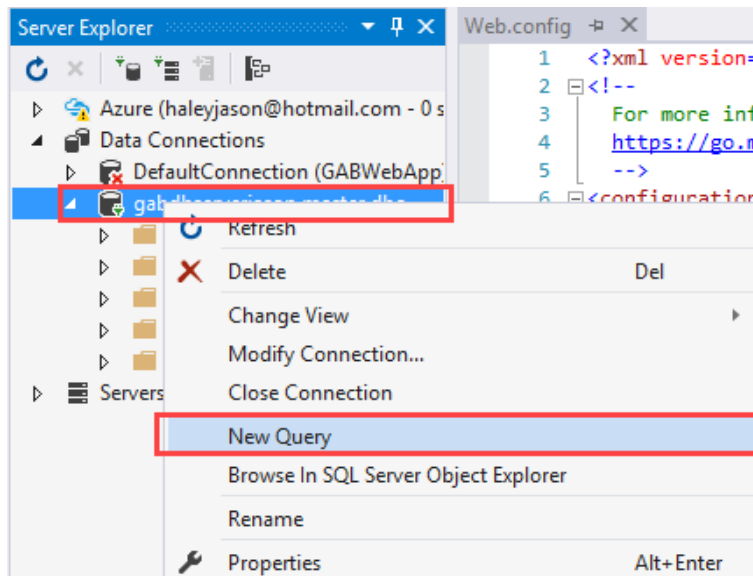
11. Click **OK**

You can now connect to the database. The next step is to add a user to the database

## Exercise 3: Add a user to the database

In this exercise, you will add a login to the master database and then add and configure a user in the database you created in exercise 1.

1. In Visual Studio, go to your **Server Explorer** and **right click** on the node for the master db that you added at the end of exercise 2 and select the **New Query** item

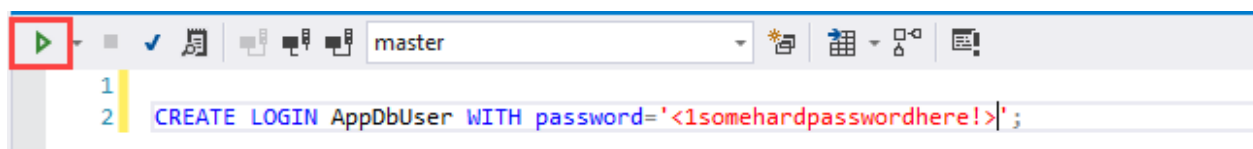


This will open a new query document in Visual Studio.

2. Type the following SQL into that query window:

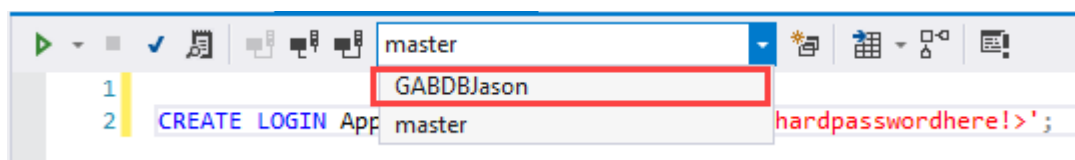
```
CREATE LOGIN AppDbUser WITH password='1somehardpasswordhere!';
```

3. Click the Execute button in the left corner



This will add the login to the master database

4. **Select the database** you created in exercise 1 from the database dropdown list



5. Remove the SQL from the query and type the following SQL in the query document:

```
CREATE USER AppDbUser FROM LOGIN AppDbUser;  
  
EXEC sp_addrolemember 'db_datareader', 'AppDbUser';
```

```
EXEC sp_addrolemember 'db_datawriter', 'AppDbUser';
EXEC sp_addrolemember 'db_ddladmin', 'AppDbUser';
```

This sql will add the user to the database and assign some roles to it.

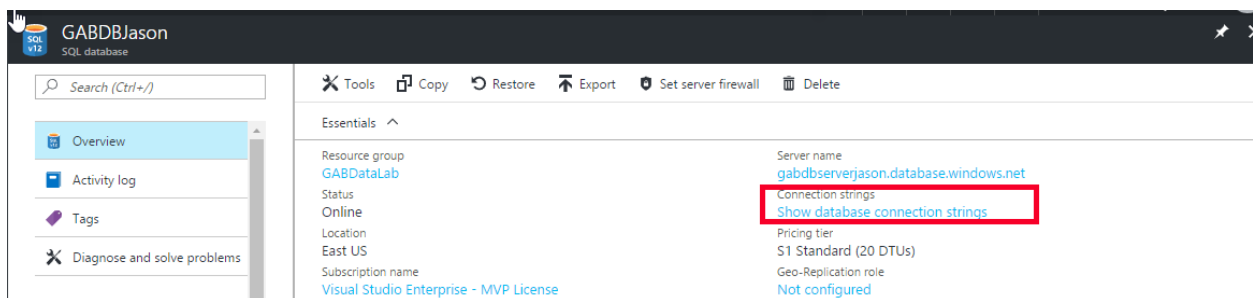
6. Execute the sql you typed above

The next step is to modify the web application to use the SQL Azure database and user you just created.

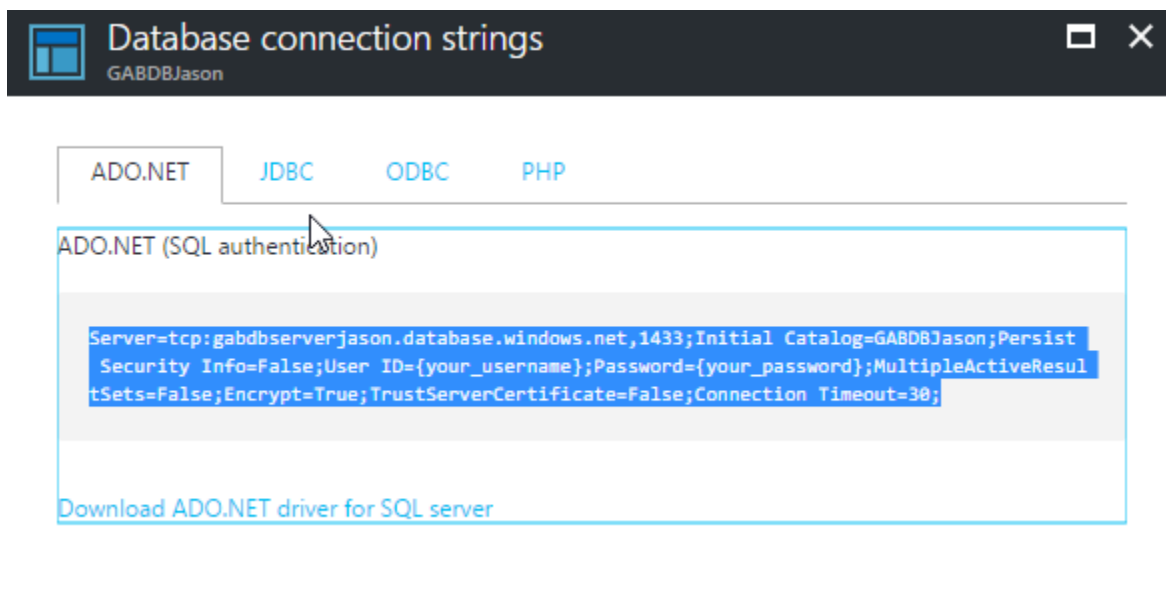
## Exercise 4: Configure Web Application to the use Azure SQL database

In this exercise, you will modify the connection string for the sample web application to use the Azure SQL database and user you created in this lab and verify it works.

1. In the Azure Portal, go to your SQL Database overview pane (if it isn't already showing)
2. On the right side of the overview pane, click on the **Show database connection strings** link



This will show the Database connection strings blade.



3. Highlight the content of the ADO.NET connection string and copy it to your clipboard
4. In Visual Studio, open the **Solution Explorer**, find the **web.config** file and open it.
5. Find the <connectionStrings> element and replace the connectionString content with the contents of your clipboard.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3 For more information on how to configure your ASP.NET application, please visit
4 https://go.microsoft.com/fwlink/?linkid=301889
5 -->
6 <configuration>
7 <configuration>
8 <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?linkid=237468 -->
9 <section name="entityframework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619344009" requirePermission="false" />
10 </configuration>
11 <connectionStrings>
12 <add name="DefaultConnection" connectionString="Server=tcp:gabdbserverjason.database.windows.net,1433;Initial Catalog=GABDBJason;Persist Security Info=False;User ID=(your_username);Password=(your_password);MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" providerName="System.Data.SqlClient" />
13 </connectionStrings>
14 </configuration>

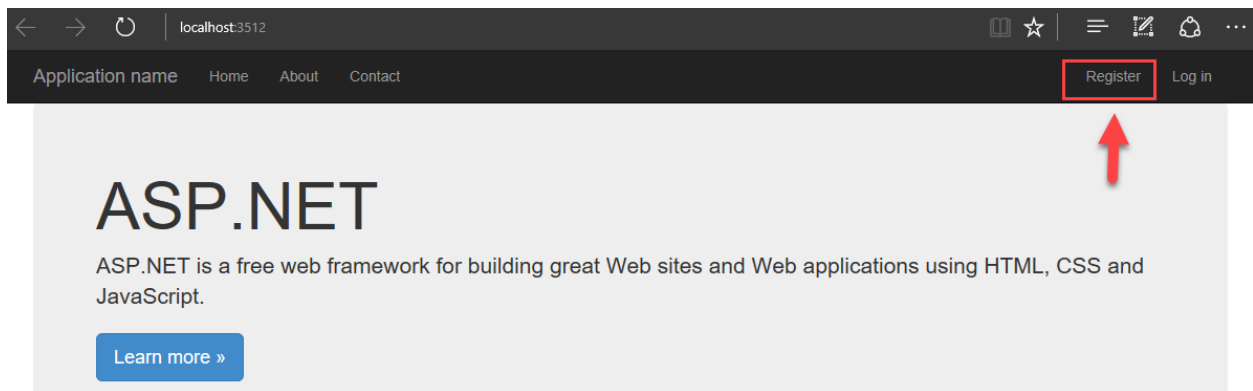
```

6. In the contents of the pasted connection string, locate the **User ID** and **Password** and replace them with AppDbUser and the password you used when setting that user up

User ID=AppDbUser;Password=1somehardpasswordhere!;

Now we can test the database connection and register a user with the web application.

7. In Visual Studio, start debugging the application (F5)
8. When the browser opens and the home page loads, click on the Register link in the upper right corner



9. Enter a email, password and confirm password

Application name
Home
About
Contact

## Register.

Create a new account.

Email

Password

Confirm password

10. Click **Register** to add the user

Once the initialization of the user tables is complete and the user is added you should be redirected to the home page showing the email address of the user in the upper right corner.

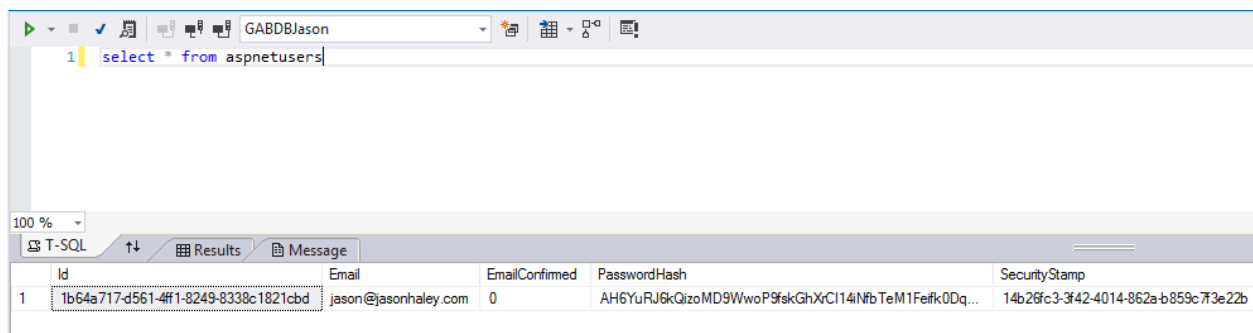
# ASP.NET

Now let's take a look at the users table.

11. In Visual Studio, find the the Sql document you used to create the users. If it is not open, then go to your Server Explorer, find the Azure database, right click and choose New Query to get another query document open.
12. Make sure your database is selected and not the master database
13. Type the following SQL into the query document and execute it:

```
select * from aspnetusers
```

This should now show a listing of the AspNetUsers table with the user you just registered.



The screenshot shows the SQL Server Enterprise Manager interface. The top toolbar includes icons for running queries, saving, and other database actions. The main query window displays the SQL statement `select * from aspnetusers`. Below the query window, the 'Results' tab is active, showing a table with one row of data. The table has five columns: Id, Email, EmailConfirmed, PasswordHash, and SecurityStamp. The data row shows a user with Id '1b64a717-d561-4ff1-8249-8338c1821cbd', Email 'jason@jasonhaley.com', EmailConfirmed '0', PasswordHash 'AH6YuRJGkQizoMD9WwoP9fGkGhXcI14INfbTeM1FeIfk0Dq...', and SecurityStamp '14b26fc3-3f42-4014-862a-b859c7f3e22b'.

	Id	Email	EmailConfirmed	PasswordHash	SecurityStamp
1	1b64a717-d561-4ff1-8249-8338c1821cbd	jason@jasonhaley.com	0	AH6YuRJGkQizoMD9WwoP9fGkGhXcI14INfbTeM1FeIfk0Dq...	14b26fc3-3f42-4014-862a-b859c7f3e22b

The sample web application is now using the Azure SQL database. You could now redeploy the web app to Azure if you wanted to see it working out there, but that is an exercise for you.