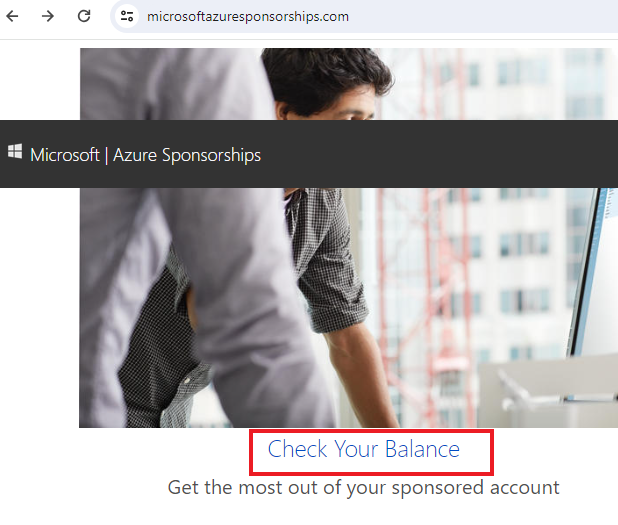


Lab 6: Create a SQL database

At the end of each lab, any resources you created in your account will be preserved. Some Azure resources, such as VM instances, may be automatically shut down, while other resources, such as storage services will be left running. Keep in mind that some Azure features cannot be stopped and can still incur charges (i.e. Azure Bastion). To minimize your costs, delete all resources and recreate them as needed to test your work during a session.

A screenshot of a computer

Description automatically generated with medium confidence



Reference: AZ-900T0X-MICROSOFTAZUREFUNDAMENTALS

# 06 - Create a SQL database

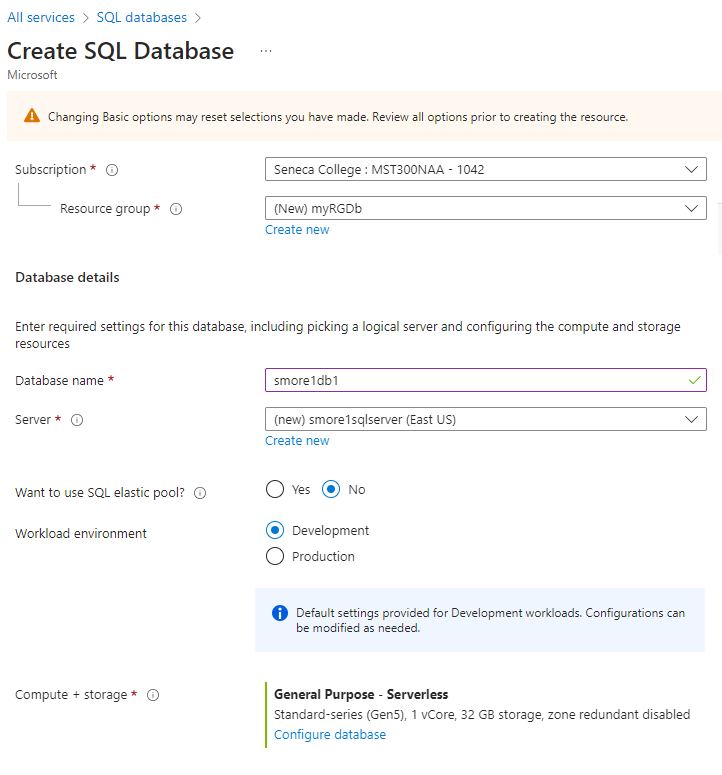
In this walkthrough, we will create a SQL database in Azure and then query the data in that database.

# Task 1: Create the database (5 min)

In this task, we will create a SQL database based on the AdventureWorksLT sample database.

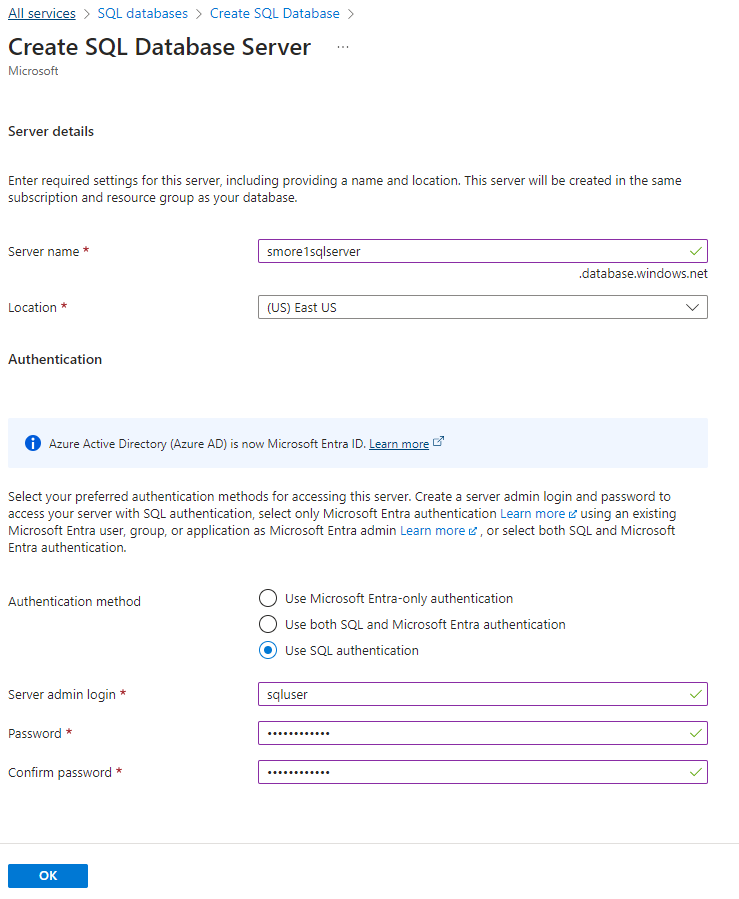
1. Sign in to the Azure portal at [**https://portal.azure.com**](https://portal.azure.com/) with your **odl\_user\_xxx** azure account
2. From the **All services** blade, search for and select **SQL databases**, and then click **+ Create**.
3. On the **Basics** tab, fill in this information.

| Setting | Value |
| --- | --- |
| Subscription | **Choose your subscription (you should see “Seneca College : <course name>”)** |
| Resource group | **myRGDb** (create new) |
| Database name | **<studentID>db1 (example: smore1db1)** |
| Want to use SQL elastic pool? | No |
| Backup storage redundancy | **Locally-redundant backup storage** |



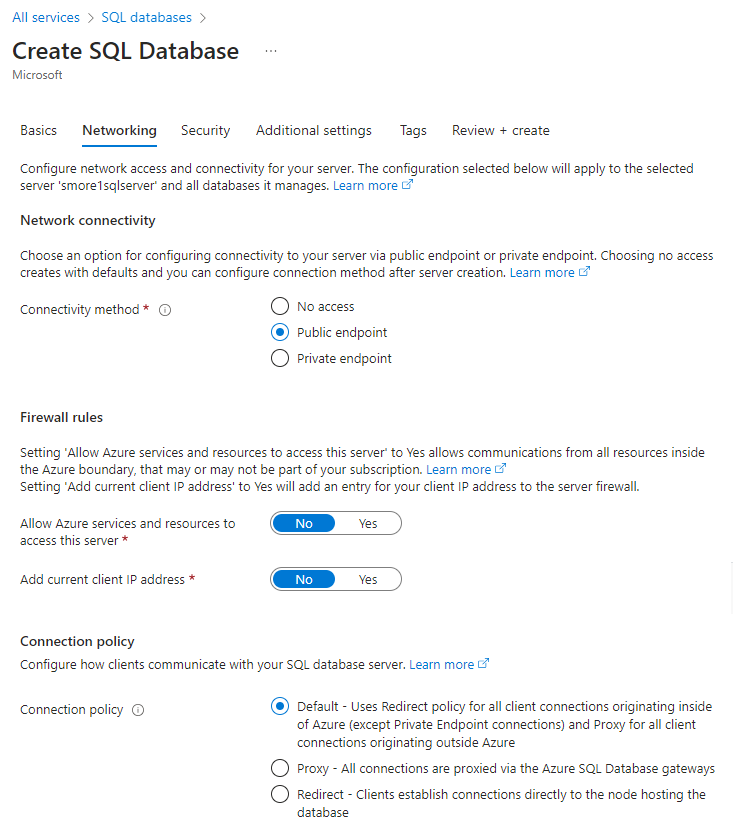
1. Next to the **Server** drop down list, click **Create new** and enter this information (replace **xxxx** in the name of the server with letters and digits such that the name is globally unique). Click **OK** when finished.

| Setting | Value |
| --- | --- |
| Server name | **<studentID>sqlserverxxxx (example: smore1sqlserver)** |
| Location | **(US) East US** |
| Authentication method | **Use SQL Authentication** |
| Server admin login | **sqluser** |
| Password | **Pa$$w0rd1234** |

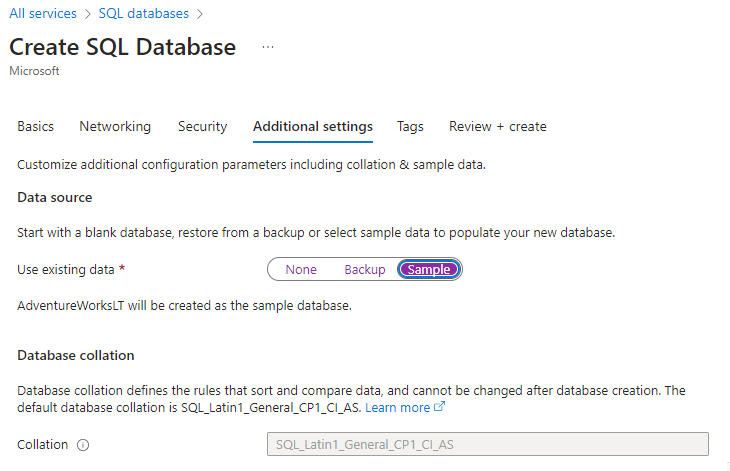


1. Move to the **Networking** tab and configure the following settings (leave others with their defaults)

| Setting | Value |
| --- | --- |
| Connectivity method | **Public endpoint** |
| Allow Azure services and resources to access this server | **Yes** |
| Add current client IP address | **No** |
|  |  |

1. 
2. Move to the **Additional settings** tab. We will be using the AdventureWorksLT sample database.

| Setting | Value |
| --- | --- |
| Use existing data | **Sample** |

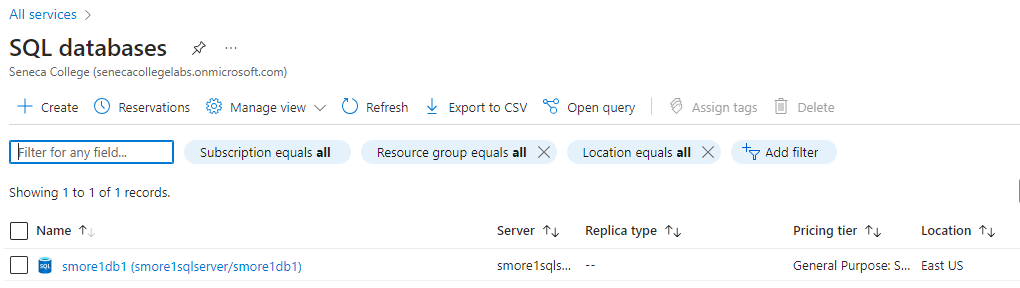


1. Click **Review + create** and then click **Create** to deploy and provision the resource group, server, and database. It can take approx. 2 to 5 minutes to deploy.
2. Go to the resource tab to locate the SQL database you created. You may need to refresh.

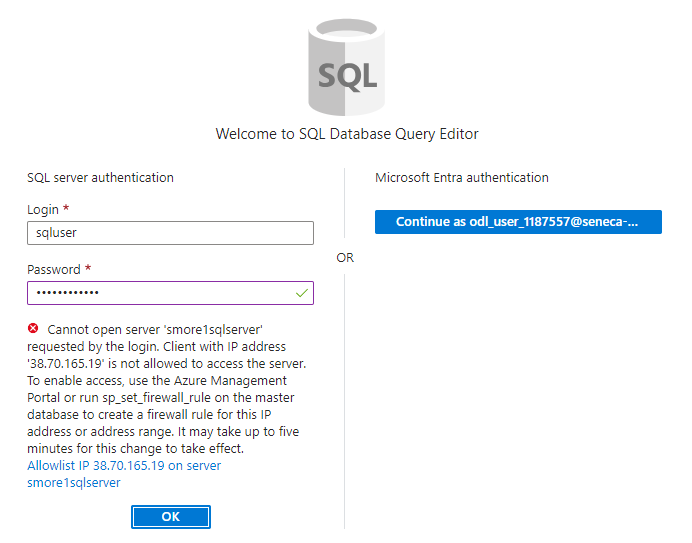
# Task 2: Test the database.

In this task, we will configure the SQL server and run a SQL query.

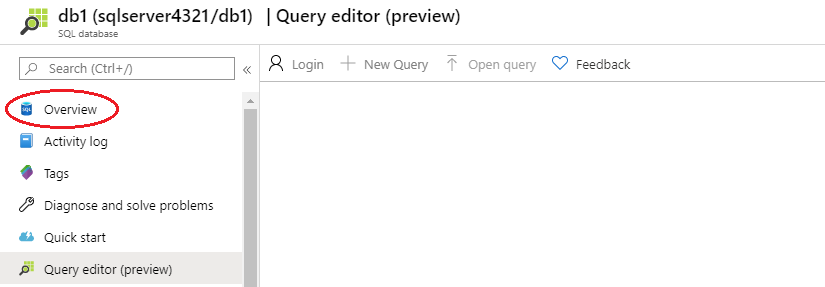
1. From the **All services** blade, search and select **SQL databases** and ensure your new database was created. You may need to **Refresh** the page.



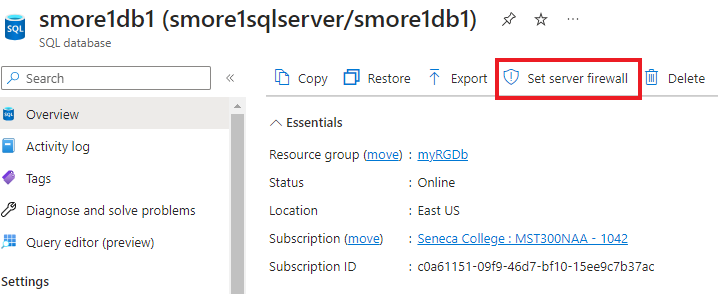
1. Click the <**your DB name**> entry representing the SQL database you created, and then click **Query editor (preview)**.
2. Login as **sqluser** with the password **Pa$$w0rd1234**.
3. You will not be able to login. Read the error closely and make note of the IP address that needs to be allowed through the firewall.

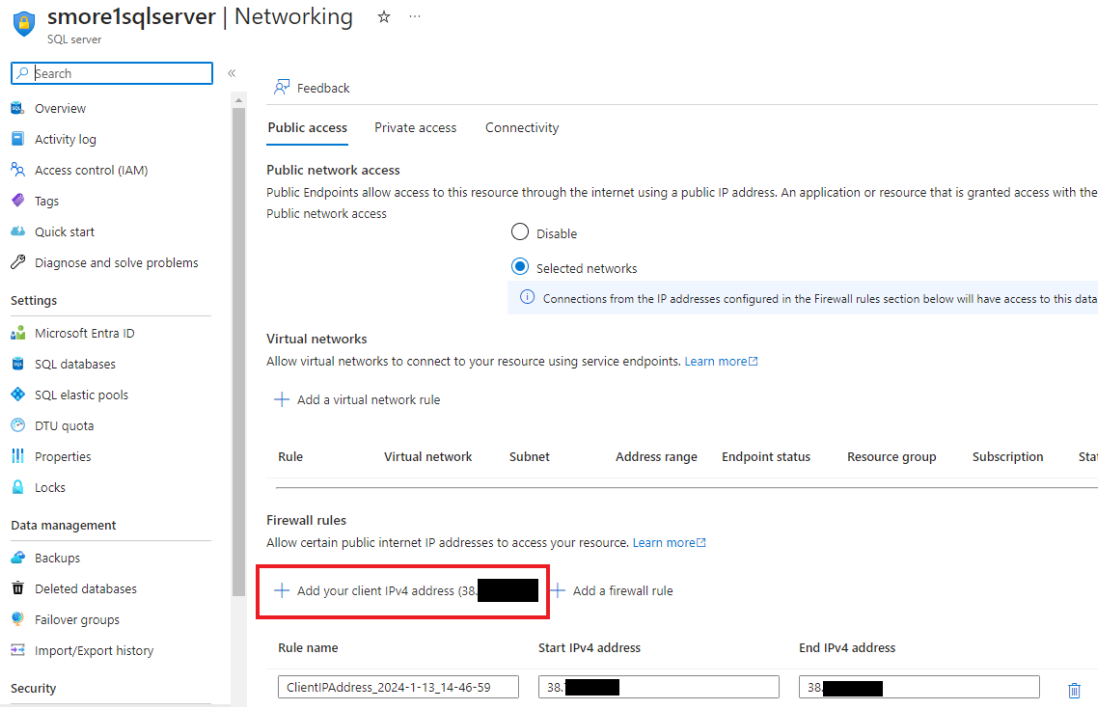


1. From < **your DB name** > blade, click **Overview**.

[](https://microsoftlearning.github.io/AZ-900T0x-MicrosoftAzureFundamentals/Instructions/images/0504.png)

1. From the SQL server **Overview** blade, select “**Set Server Firewall**” , under Firewall Rules click + **Add your client IPv4 address (x.x.x.x).** Be sure to **Save** your changes.





1. Return to your SQL database and the **Query Editor (Preview)** login page. Try to login again as **sqluser** with the password **Pa$$w0rd1234**. This time you should succeed. Note that it may take a couple of minutes for the new firewall rule to be deployed.
2. Once you log in successfully the query pane appears, enter the following query into the editor pane and click **run**.

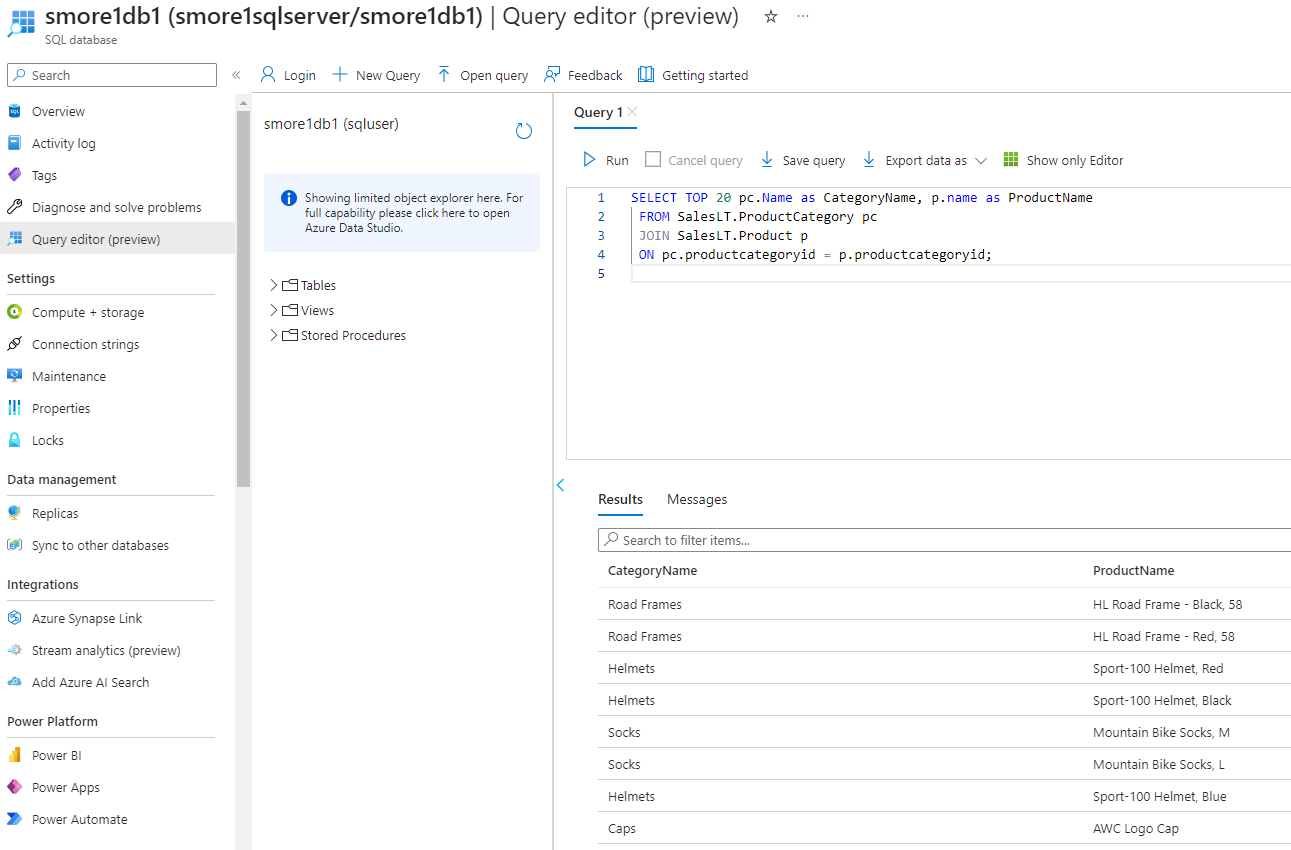
CodeCopy

SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName

FROM SalesLT.ProductCategory pc

JOIN SalesLT.Product p

ON pc.productcategoryid = p.productcategoryid;



Congratulations! You have created a SQL database in Azure and successfully queried the data in that database.

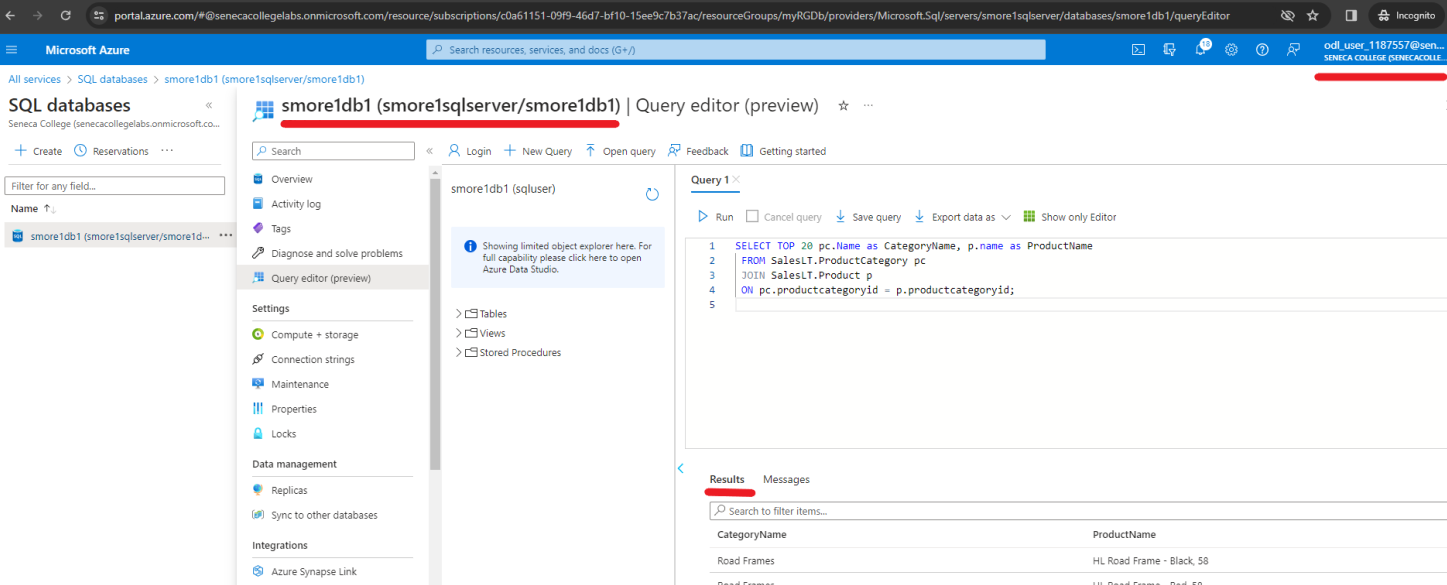
**Note**: To avoid additional costs, you can remove all resources in the resource group. Search for resource groups, click your resource group, and then delete the resources within the resource group. **DO NOT DELETE YOUR RESOURCE GROUP.**

# Submission Requirements

Submit a screenshot with the following information:

**Screenshot #1:**

* A successful query of sample data from your SQL database
* The Azure Portal with your **CloudLab Account** [requires another browser window]
  + **Note**: underline the above items as described in the below picture



**Screenshot #2:**

* Successful deletion of all resources within resource group. **DO NOT DELETE YOUR RESOURCE GROUP!**
  + To delete all resources with a resource group, go to “**Resource Group**”, select “**myRGDb** ”, select all resources within the resource group, and select “**Delete**”

