# Lab Answer Key: Module 7: Planning and implementing Azure SQL Database

# Lab: Planning and implementing Azure SQL Database

## Exercise 1: Creating, securing, and monitoring an Azure SQL Database

#### Task 1: Create an Azure SQL Database

1. Sign in to the MIA-CL1 lab virtual machine as **Student** with the password **Pa$$w0rd**.
2. Start Internet Explorer, browse to **https://portal.azure.com**, and sign in by using the Microsoft account that is the Subscription Administrator or Co-Administrator of your Azure subscription.
3. In the Hub menu on the left, click **New**, click **Databases**, and then click **SQL Database**.
4. On the **SQL database** blade, in the **Database name** box, type **operations**.
5. In the **Resource Group** section, locate the **Create new** radio button and select it.
6. In the **Resource Group** section, in the text box, enter **OpsRG** as the name of the *Resource Group*.
7. Ensure that **Blank database** is selected in the Select source drop down list.
8. Click **Server**, and then on the **Server** blade, click **Create a new server**.
9. On the **New server** blade, enter the following settings and then click **Select**:

* Server name: any valid unique name
* Server admin login: **Student**
* Password: **Pa$$w0rd**
* Confirm password: **Pa$$w0rd**
* Location: the closest Azure region (to your location)
* Create V12 server (Latest update): **Yes**
* Allow azure services to access server: Enabled

1. Click **Pricing tier**.
2. On the **Choose your pricing tier** blade, select **S1 Standard** and click Select.
3. On the **SQL database** blade, ensure that **Pin to dashboard** is selected and click **Create**. Then wait for the SQL Database to be created.
4. After the database is created, the portal will automatically display its **Settings** blade.

#### Task 2: Configure server firewall rules

1. Navigate to the **operations** blade in the Azure portal in the Internet Explorer window.
2. On the **operations** blade, click the hyperlink containing the server name.
3. Navigate to the **Settings** blade of the server and click **Show firewall settings**.
4. On the **Firewall settings** blade, note the value of the **Client IP address** entry.
5. On the **Firewall settings** blade, specify the following:

* RULE NAME: **AllowLabVM**
* START IP: \*\*\_XXX.XXX.\_0.0\*\*
* END IP: ***XXX.XXX*.255.255**

**Note:** where ***XXX.XXX*** represents the first two octets of the value of the **Client IP address** entry.

1. Click **Save**

#### Task 3: Use SQL Server Management Studio

1. Start SQL Server 2014 Management Studio, and in the **Connect to Server** dialog box, specify the following settings (replacing server\_name with the unique name you specified when creating your SQL Database server), and click **Connect**:

* Server type: **Database Engine**
* Server name: **server\_name.database.windows.net**
* Authentication: **SQL Server Authentication**
* Login: **Student**
* Password: **Pa$$w0rd**

1. In SQL Server Management Studio, in the Object Explorer, under the server name expand **Databases**, and verify that the **operations** database is listed.
2. In SQL Server Management Studio, navigate to the D:07folder, open the **Operations.sql** file and view the Transact-SQL code it contains.
3. On the toolbar, in the **Available Databases** list, select **operations**. Click **Execute**
4. Click **New Query** and enter the following Transact-SQL code in the new query pane:

SELECT \* FROM dbo.serverlist;

1. On the toolbar, in the **Available Databases** list, ensure that **operations** is selected. Then click **Execute**.
2. View the query results and verify that a list of three servers and their IP addresses is returned.
3. Keep SQL Server Management Studio and Internet Explorer open.

#### Task 4: View database metrics

1. In Internet Explorer, in the Azure portal, navigate to the **operations** SQL Database blade.
2. On the **operations** blade, note the charts displayed in the **Monitoring** section, which show resource utilization in terms of DTU percentage.
3. Click **Edit**, in the **Resource utilization** chart, deselect **DTU percentage** and click **Total database size**, and then click **OK**.
4. Click the chart. This will display the **Metric** blade.
5. On the **Metric** blade, click **Add alert**. Then, on the **Add an alert rule** blade, specify the following settings and click **OK**:

* Resource: leave the default setting in place
* Name: **operations storage alert**
* Description: **storage alert for** **operations database**
* Metric: **total database size**
* Condition: **greater than**
* Threshold: **1024**
* Period: **over the last 5 minutes**
* Email owners, contributors, and readers: selected
* Additional administrator email(s): any email address
* Webhook: leave blank

1. Keep Internet Explorer open for the next exercise

**Result**: After completing this exercise, you should have created an Azure SQL Database named operations on a new server with a name of your choosing. You should also have used SQL Server Management Studio to create a table named dbo.serverlist and created an alert to help you monitor database storage.

## Exercise 2: Migrating a Microsoft SQL Server Database to Azure SQL Database

#### Task 1: Deploy a database to Azure

1. In SQL Server Management Studio, in Object Explorer, in the **Connect** drop-down list, click Database Engine.
2. In the **Connect to Server** dialog box, specify the following settings, and click **Connect**:

* Server type: **Database Engine**
* Server name: **MIA-CL1**
* Authentication: **Windows Authentication**

1. In SQL Server Management Studio, in Object Explorer, under the **MIA-CL1** server, expand **Databases** and verify that the **sales** database is listed.
2. Right-click the **sales** database, point to **Tasks**, and then click **Deploy Database to Windows Azure SQL Database**.
3. In the **Deploy Database "sales"** wizard, on the **Introduction** page, click **Next**.
4. On the **Deployment Settings** page, click **Connect**. Then in the **Connect to Server** dialog box, specify the following settings (replacing ***server\_name*** with the unique name of your SQL Database server) and click **Connect**:

* Server type: **Database Engine**
* Server name: ***server\_name*.database.windows.net**
* Authentication: **SQL Server Authentication**
* Login: **Student**
* Password: **Pa$$w0rd**

1. On the **Deployment Settings** page, ensure that the new database name is **sales** and note the temporary file name used for the .bacpac file that will be exported and imported, ensure that the **Service Objective** **is set** to **S2,** and then click **Next**..
2. On the **Summary** page, click **Finish**.
3. On the **Results** page, verify that the operation completed successfully, and click **Close**.
4. In SQL Server Management Studio, in Object Explorer, if necessary, right-click the **Databases** folder under your Azure SQL Database server and click **Refresh** to verify that the sales database has been copied to this server.

#### Task 2: Configure SQL Database security

1. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, expand **Security**, expand **Logins**, and verify that only the **Student** login is listed.
2. Right-click **Logins** and click **New Login**. Then, replace the auto-generated Transact-SQL script that is generated as shown here and then click **Execute**:

CREATE LOGIN SalesApp WITH PASSWORD = 'Pa$$w0rd' GO

1. In Object Explorer, right-click the **Logins** folder and click **Refresh** to verify that the **SalesApp** login has been created.
2. In Object Explorer, in the **Databases** folder for your Azure SQL Database server, expand the **sales** database, expand **Security**, and expand **Users**.
3. Right-click **Users** and click **New User**. Then, modify the Transact-SQL script that is generated as shown below and then click **Execute**:

CREATE USER SalesApp FOR LOGIN SalesApp WITH DEFAULT\_SCHEMA = dbo GO EXEC sp\_addrolemember 'db\_owner', 'SalesApp' GO

1. In Object Explorer, right-click the **Users** folder and click **Refresh** to verify that the **SalesApp** user has been created.
2. Keep SQL Server Management Studio open for the next exercise.

#### Task 3: Configure an application connection string

1. Start Visual Studio, navigate to the **D:07\* folder and open the** SalesApp.sln\*\* solution.
2. In Solution Explorer, double-click **Web.config**.
3. In Web.config, note that the **SalesConnectionString** element contains a **connectionString** attribute that connects to the **sales** database on the **localhost** server using integrated security (Windows authentication).
4. In Internet Explorer, on the tab containing the preview Azure portal, on the Hub menu, click **Browse** and then click **SQL databases**.
5. On the **SQL databases** blade, click the **sales** database.
6. On the **sales** blade, under Essentials, click **Show database connection strings**.
7. On the **Database connection strings** blade, click the **Click to copy** icon for the **ADO.NET** connection string. If prompted, click **Allow access**.
8. Minimize Internet Explorer, you will return to it in the next exercise.
9. In Visual Studio, in Web.config, select the existing value for the **connectionString** attribute and then paste the connection string you copied to replace it.
10. In the pasted connection string, set the value of the **User ID** parameter to **SalesApp@\_server\_name\_** (where ***server\_name*** is the unique name of your Azure SQL Database server). Next, set the value of the Password parameter to **Pa$$w0rd** (by replacing the **{your\_password\_here}** placeholder. The new connectionString value should look similar to this (on a single line):

Server=tcp:server\_name.database.windows.net, 1433;Database=sales; User ID=SalesApp@server\_name; Password=Pa$$w0rd;Encrypt=True; TrustServerCertificate=False; Connection Timeout=30;

1. Save Web.config. Then on the **Debug** menu, click **Start Debugging**.
2. When Internet Explorer opens, verify that the sales application shows invoice history data for the selected customer. The data is retrieved from the sales database you migrated to Microsoft Azure SQL Database.
3. Close the Internet Explorer window that contains the **Customer Invoice History** page, ensure that Visual Studio debugger is stopped, and then close Visual Studio, saving changes if prompted.

**Result**: After completing this exercise, you should have deployed the sales SQL Server database on the local SQL Server instance to your Azure SQL Database server, and configured the SalesApp web application to use a connection string for the new Azure SQL Database.

## Exercise 3: Restoring a database

#### Task 1: Delete a database

1. In Internet Explorer, in the Azure Portal, in the Hub menu, click **More services**, and then click **SQL Database**.
2. On the **SQL Databases** blade, click the **operations** database.
3. On the **operations** blade, click **Restore**.
4. On the **Restore** blade, verify whether a restore point is available. If not, wait until that is the case.
5. On the **operations** blade, click **Delete**.
6. When prompted to confirm, click **Yes**.
7. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, right-click the **Databases** folder and click **Refresh** to verify that the **operations** database is no longer on the server.

#### Task 2: Restore a deleted database

1. In the Hub menu of the Azure Portal, click **Browse**, select **SQL Servers**, and click the name of the SQL server where the **operations** database was created.
2. On the **SQL server** blade, scroll down to the **Operations** section and click **Deleted databases**.
3. On the **Deleted databases** blade, click **operations**.
4. On the **Restore** blade, set the database name to **operations**. Notice that you are restoring the most recent restore point to the same server.
5. Click **OK**.
6. Wait for the restore operation to complete by monitoring **Notifications** area in the portal or on the **Audit Logs** blade (this can take several minutes). > **Note:** If the initial restore attempt fails, try again.
7. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, right-click the **Databases** folder and click **Refresh** to verify that the operations database has been restored.
8. In SQL Server Management Studio, click **New Query** and enter the following Transact-SQL code in the new query pane:

SELECT \* FROM dbo.serverlist

1. On the toolbar, in the **AvailableDatabases** list, ensure that **operations** is selected and then click **Execute**.
2. View the query results and verify that a list of three servers and their IP addresses is returned.

#### Task 3: Reset the environment

1. Close all open applications without saving any files.
2. On the taskbar, right-click **Windows PowerShell**, and then click **Run as administrator**. In the **User Account Control** dialog box, click **Yes**.
3. Type the following command, and then press Enter:

Reset-Azure

1. When prompted (twice), sign in using the Microsoft account associated with your Azure subscription.
2. If you have multiple Azure subscriptions, select the one you want to target by the script.
3. When prompted for confirmation, type **y**.

**Note:** This script will remove Azure services in your subscription. We, therefore, recommend that you use an Azure trial pass that was provisioned specifically for this course, and not your own Azure account. The script will take 5-10 minutes to reset your Microsoft Azure environment, ready for the next lab. The script removes all storage, VMs, virtual networks, cloud services, and resource groups

**Important**: The script might not be able to get exclusive access to a storage account to delete it (if this occurs, you will see an error). If you find objects remaining after the reset script is complete, you can re-run the **Reset-Azure** script, or use the Azure portal and Azure classic portal to manually delete all the objects in your Azure subscriptionâ€”with the exception of the default directory.

#### Task 4: To prepare for the next module

Leave the virtual machines running for the next module.

**Result**: After completing this exercise, you should have deleted and restored the operations database.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/legalcode), additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.