# Module 7: Planning and implementing Azure SQL Database

# Lab: Planning and implementing Azure SQL Database

### Scenario

Managers at A. Datum are planning to migrate some of the company's application databases to the cloud.

To achieve this goal, you plan to use Microsoft Azure SQL Database. You have been asked to test SQL Database by creating a new database of A. Datum servers and by migrating sample data from the A. Datum customer relationship management system. Managers have asked you to investigate how SQL Database will support an existing custom application used with A. Datum, as well as disaster recovery features.

### Objectives

After completing this lab, you will be able to:

* Provision Azure SQL Database.
* Migrate a SQL Server database to Azure SQL Database.
* Restore a deleted database.

### Lab Setup

Estimated Time: 60 minutes

Virtual Machine: \*\* 20533C-MIA-CL1\*\*

User Name: \*\* Student\*\*

Password: \*\* Pa$$w0rd\*\*

Before starting this lab, ensure that you have performed the "Preparing the Environment" demonstration tasks at the beginning of the first lesson in this module, and that the setup script has completed.

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

### Scenario

The operations team at A. Datum currently uses a Microsoft SQL Server database to store inventory of company's servers. You want to investigate the option of using Azure SQL Database to host this database. The operations team is interested in monitoring the performance of this database in Azure.

Note: The Microsoft Azure portal is continually improved, and the user interface might have been updated since this lab was written. Your instructor will make you aware of any differences between the steps described in the lab and the current Azure portal user interface.

The main tasks for this exercise are as follows:

1. Create an Azure SQL Database
2. Configure server firewall rules
3. Use SQL Server Management Studio
4. View database metrics

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

1. Sign in to the 20533C-MIA-CL1 lab virtual machine as **Student** with the password **Pa$$w0rd**
2. Launch Internet Explorer and sign in to the Azure portal by using the Microsoft account that is the Subscription Administrator or Co-Administrator of your Azure subscription.
3. From the Azure portal, create a new SQL database with the following parameters:

* Name **: operations**
* SQL server

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

* Select source: **Blank database**
* Pricing tier: **S1 Standard**
* Resource group: **OpsRG**
* Pin to dashboard: Enabled

1. After the database is created, the portal will automatically display its **Settings** blade.

#### Task 2: Configure server firewall rules

1. From the Azure portal, navigate to the Firewall settings blade of **operations** database.
2. On the **Firewall settings** blade, identify the public IP address corresponding to your lab virtual machine.
3. On the **Firewall settings** blade, create a new rule with the following settings:

* 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 client IP address.

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

1. Use SQL Server Management Studio to connect to the Azure SQL server with the following settings:

* 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 Object Explorer, verify that the **operations** database is listed
2. From SQL Server Management Studio, execute the Transact SQL script stored in the **Operations.sql** file in the D:07folder.
3. From SQL Server Management Studio, run the following query against the **operations** database:

SELECT \* FROM dbo.serverlist;

1. View the query results and verify that a list of three servers and their IP addresses is returned.
2. Keep SQL Server Management Studio and Internet Explorer open.

#### Task 4: View database metrics

1. From the Azure portal, in Internet Explorer window, 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. Add **Total database size** to the Resource utilization chart.
4. Display the **Metric** blade for the **Resource utilization** chart.
5. On the **Metric** blade, add an alert with the following settings:

* 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

### Scenario

The sales team at A. Datum uses a customer relationship management system application to track customer invoices. The application currently stores customer data in an on-premises Microsoft SQL Server database. You want to demonstrate that Azure can support this customer relationship management system application by migrating the database for this application to Azure SQL Database, and then reconfiguring the application to use the new, cloud-based database.

The main tasks for this exercise are as follows:

1. Deploy a database to Azure
2. Configure SQL Database security
3. Configure an application connection string

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

1. In SQL Server Management Studio, connect to the SQL Server instance running on **MIA-CL1** by using Windows authentication.
2. Verify that the **sales** database is listed in the **Databases** folder for the **MIA-CL1** server.
3. Right-click the **sales** database, point to **Tasks**, and click **Deploy Database to Windows Azure SQL Database**. Then use the wizard to deploy the **sales** database from **MIA-CL1** to your Microsoft Azure SQL Database server. Ensure that the **Service Objective** **is set** to **S2**.

#### 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. Create a new login named **SalesApp** with the password **Pa$$w0rd** by executing the following Transact-SQL code in the master database:

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

1. In Object Explorer, in the **Databases** folder for your Azure SQL Database server, expand the **sales** database, expand **Security**, and expand **Users** to view the users that are defined in the sales database.
2. Create a user named **SalesApp** for the **SalesApp** login. The user should have a default schema of **dbo**, and should be added to the **db\_owner** database role. You can create the user by executing the following Transact-SQL code in the **sales** database:

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

1. Keep SQL Server Management Studio open for the next exercise.

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

1. Start Visual Studio and open the **SalesApp.sln** solution in the **D:07folder**. Then, open its **Web.config** file and note that the **SalesConnectionString setting** connects to the sales database on the localhost server using integrated security (Windows authentication).
2. In Internet Explorer, in the Azure portal, browse to the **sales** database blade.
3. On the **sales** database blade, identify the value of the **ADO.NET** database connection string and copy it to the Clipboard.
4. In Visual Studio, replace the existing connection string with the one you copied from the Azure portal. Then in the copied connection string, set the value of the User ID parameter to **SalesApp@ *server\_name*** (where ***server\_name*** is the name of your Azure SQL Database server). Next, set the value of the Password parameter to **Pa$$w0rd**. The new connectionString value should look similar to this:

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 sales application, ensure that the 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

### Scenario

The operations database you created is considered a mission-critical source of data for IT employees at A. Datum. Before business decision makers can commit to using Azure to host this database, you must ensure that the database can be recovered in the event of accidental deletion.

The main tasks for this exercise are as follows:

1. Delete a database
2. Restore a deleted database
3. Reset the environment
4. To prepare for the next module

#### Task 1: Delete a database

1. In Internet Explorer, in the Azure portal, from the **operations** SQL Database blade, verify that there is at least one restore point. If not, wait until that is the case.
2. After you verified that there is a restore point, delete the **operations** SQL Database.
3. In SQL Server Management Studio, refresh the **Databases** folder for your Azure SQL Database server to verify that the **operations** database is no longer on the server.

#### Task 2: Restore a deleted database

1. In the Azure portal, browse to the SQL server where **operations** database was created.
2. From the SQL server blade, display the list of its deleted databases.
3. From the **Deleted databases** blade, initiate the restore of **operations** database.
4. 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.
5. In SQL Server Management Studio, in Object Explorer, refresh the list of databases to verify that the **operations** database has been restored.
6. In SQL Server Management Studio, run the following query against the **operations** database:

SELECT \* FROM dbo.serverlist

1. View the query results and verify that a list of three servers and their IP addresses is returned.

#### Task 3: Reset the environment

1. Launch **Windows PowerShell** as Administrator
2. From the Windows PowerShell prompt, run:

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.

**Question** If the SalesApp web application was deployed to a server with a fixed public IP address, how could you enable it to access the sales Azure SQL Database without allowing it to access any other Azure SQL Database on the same server running SQL?

©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.