

# Azure SQL Database Essentials

---

Welcome [FIRSTNAME]

---

## Labs

---

[M01L02Lab01 - Create an Azure SQL Database](#)

[M02L01Lab01 - Various Tools to Manage Azure SQL Database](#)

[M02L02Lab01 - Scale up or down an Azure SQL Database](#)

[M02L03Lab01 - Elastic Database Jobs](#)

[M03L01Lab01 - Point in time restore of an Azure SQL Database](#)

[M03L02Lab01 - Configure Geo Replication for an Azure SQL Database](#)

[M04L06Lab01 - Implement Always Encrypted](#)

[M04L07Lab01 - Implement Row Level Security](#)

[M04L10Lab01 - Vulnerability Assessment](#)

[M05L02Lab01 - Monitoring Query Performance using Query Performance Insights](#)

[M05L05Lab01 - Monitoring Azure SQL Database Performance using Extended Events](#)

[M06L05Lab01 - Migrate a Compatible SQL Server Database to Azure SQL Database](#)

[M06L06Lab01 - Azure SQL Data Sync](#)

# Create an Azure SQL Database

---

## Introduction

In this lab, you will learn how to create Azure SQL Database.

## Estimated Time

20 minutes

## Objectives

After completing this lab, you will be able to:

- Create a logical SQL Server
- Configure Firewall Rules
- Create an Azure SQL Database

1. Connect to the lab virtual machine named **@lab.VirtualMachine(AzureSQLDB).SelectLink** as **[T] @lab.VirtualMachine(AzureSQLDB).Username** using **[T] @lab.VirtualMachine(AzureSQLDB).Password** as the password.
- Click the **[T] Type Text** icon to enter the associated text into the virtual machine.
2. Change the screen resolution if required.

- You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Create a new SQL Server (logical server)

---

The first step in creating an Azure SQL Database is creating an "Azure SQL Server". In this exercise, you're going to create your first Azure SQL Server.

---

## Tasks

- You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

- 1. Connect to Microsoft Azure Portal.

From within the lab environment, open a web browser, and in the navigation bar type <https://portal.azure.com> to connect to Microsoft Azure Portal. (This may be the default home page). Login as `@lab.CloudPortalCredential(LabUser).Username` using `@lab.CloudPortalCredential(LabUser).Password` as the password.

- 2. If the **Save password** dialog box appears, click **Save**, otherwise continue to next step.
- 3. If the **More information required** dialog box appears, click **Ask later**, otherwise continue to next step.
- 4. If the **Stay signed in?** dialog box appears, click **Yes**, otherwise continue to next step.
- 5. If the **Welcome to Microsoft Azure** dialog box appears, click **Maybe later**, otherwise continue to next step.
- 6. Click **Create a resource**.



- 7. In the search field type **Azure SQL** and press **Enter**.



- 8. If the **Private Offers** dialog box appears, click **Close**, otherwise continue to next step.
- 9. In the result list select **Azure SQL** and then click on the **Create** button.
- 10. On the **Select SQL deployment option** page, under the **SQL Database** window, Select **Database server** as the Resource Type. Click **Create** to create the SQL Server (logical server).

## Select SQL deployment option

Microsoft

Feedback

### How do you plan to use the service?

**SQL databases**  
Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type

Database server

Single database

Elastic pool

Database server

- 11. Configure the SQL Server (logical server)

Specify the following options of your logical server.

Option	Value
<b>Subscription</b>	Leave default
<b>Resource Group</b>	Leave default
<b>Server Name</b>	Specify a unique server name
<b>Location</b>	Select the location that is the closest to your location
<b>Authentication Mode</b>	<b>SQL Authentication</b>
<b>Server Admin Login</b>	<input type="text" value="AzureAdmin"/> AzureAdmin
<b>Password</b>	<input type="text" value="Passw0rd1"/> Passw0rd1
<b>Confirm password</b>	<input type="text" value="Passw0rd1"/> Passw0rd1

**i** Leave other options to default

- 12. Click the **Review + Create** button.
- 13. Observe the **Review** page and then click the **Create** button.
- 14. Check the details of your new logical SQL Server by clicking on the **Notifications** icon (a bell icon) at the top right of the portal. Wait until the **Deployment succeeded** message appears.



## Notifications

X

More events in the activity log →

Dismiss all ▾

✓ Deployment succeeded

X

Deployment

'Microsoft.SQLServer.createServer\_63f1ce01b5af4efa9ee20e1bcde2df4' to resource group 'AzureDB-RG-lod16776558' was successful.

**Go to resource**

**Pin to dashboard**

a few seconds ago

- 15. Once deployment has completed, navigate to the logical SQL server that you created by clicking on the **Go to resource** button.

[!NOTE] Alternatively, you can click **All Services** on the left side navigation blade then enter **T sql servers** in the search field and click **SQL servers** from the list, then click on your SQL server name to open the SQL server blade.

► ! [M01 L02 Lab01 1](Media/M01L02Lab01\_M01L02Lab01\_9.png)

- 1. Record the unique name of your logical server in the following text box:

+ Create database + New elastic pool + New dedicated SQL pool (formerly SQL DW) ...

^ Essentials

Resource group ( <a href="#">move</a> ) <a href="#">AzureDB-RG-lod33268965</a>	Server admin AzureAdmin
Status Available	Networking <a href="#">Show networking settings</a>
Location East US	Active Directory admin <a href="#">Not configured</a>
Subscription ( <a href="#">move</a> ) <a href="#">ASD Cloud Slice 5</a>	Server name Unique Server Name <input type="text"/> database.windows.net

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Create firewall rules

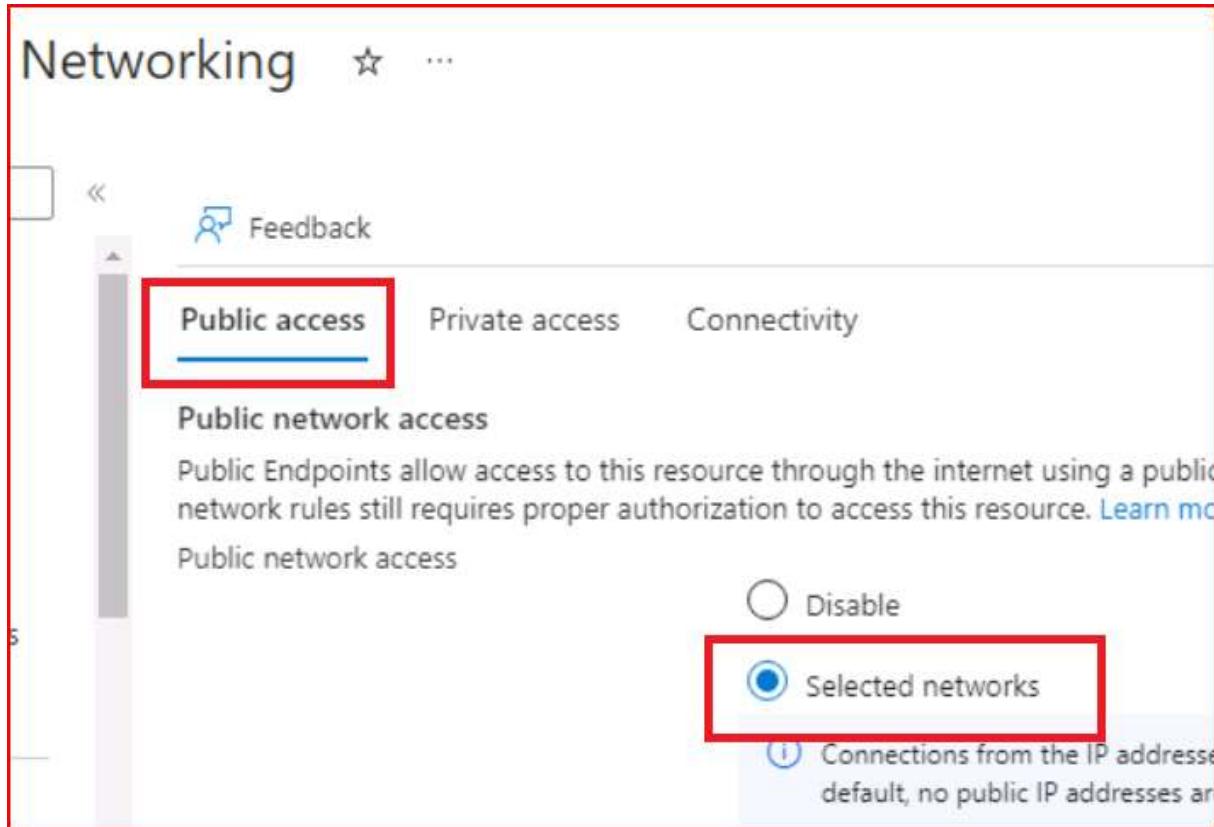
By default, nobody can connect to the logical server that you have created. To allow connections, you need to configure the SQL Server or Database Firewall rules

### Tasks

- 1. On the right side of the **Overview** page for your logical server, navigate to the firewall settings by clicking the link **Show networking settings**.

Server admin  
AzureAdmin  
  
Networking  
[Show networking settings](#)

- 2. On the **Networking** page of your logical server, select the **Public access** tab and then click on the **Selected networks** radio button.



- 3. Scroll down the **Networking** page, click on the **Add your Client IP4 address** button, and then click **Save**.

## Firewall rules

Allow certain public internet IP addresses to access your resource. [Learn more](#)

 Add your client IPv4 address

 Add a firewall rule

[!NOTE]Leave remaining Firewall options to default

- 1. On the **Success!** dialog box click **OK**.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 3: Create an Azure SQL Database

---

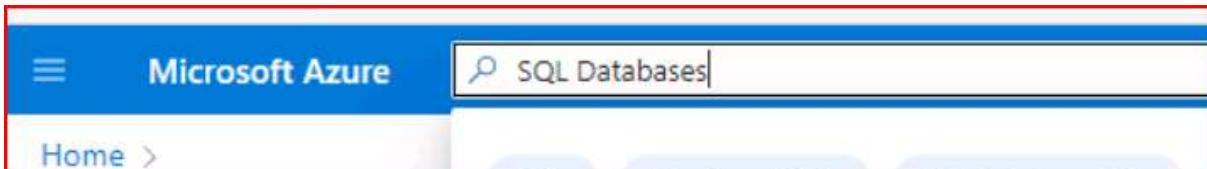
After creating the logical SQL Server and the firewall rule, we can create the database

---

### Tasks

- 1. Find SQL Databases

In the top search bar, fill in **T SQL Databases** and press **Enter**



- 2. Under SQL Databases, click the **Create** button. If tour navigation dialog boxes appears, close them.

A screenshot of the "SQL databases" blade in the Azure portal. At the top, it says "SQL databases" and shows a connection status. Below that, it says "ASD Cloudslice (lodsasdoutlook.onmicrosoft.com)". There are three buttons: "+ Create", "Reservations", and "Manage view".

- 3. Configure the SQL Database

Specify the following options:

Setting	Value
<b>Subscription</b>	Leave default
<b>Resource Group</b>	Leave default
<b>Database name</b>	<b>T Salesdb</b>
<b>Server</b>	(The name of the logical server you have created in Exercise 1)
<b>Want to use SQL elastic pool?</b>	No
<b>Workload environment</b>	Development

Setting	Value
<b>Compute + storage</b>	Click the link <b>Configure database</b> . In the <b>Service tier</b> drop down list under the <b>DTU-based purchasing model</b> , select <b>Basic</b> , then click <b>Apply</b> .
<b>Backup storage redundancy</b>	If this option shows to you, choose <b>Geo-redundant backup storage</b> .

Home > SQL databases >

## Create SQL Database

Microsoft

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

Salesdb

Server \* ⓘ

azserv (Brazil South)

[Create new](#)

Want to use SQL elastic pool? \* ⓘ

Yes  No

Compute + storage \* ⓘ

Basic

2 GB storage

[Configure database](#)

### Backup storage redundancy

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy ⓘ

Locally-redundant backup storage - Preview

Zone-redundant backup storage - Preview

Geo-redundant backup storage

⚠ Selected value for backup storage redundancy is Geo-redundant backup storage. Note that database backups will be geo-replicated to the paired region. [Learn more ↗](#)

💡 Your use of either of the [Preview backup storage redundancy options](#)

[Review + create](#)

[Next : Networking >](#)

- 4. Leave any other options to default, and then click **Review + create**.
- 5. Click **Create** to create new database.
- 6. Examine the Database

Once the deployment has succeeded, check the details of your new database by clicking on the **Notifications** icon (a bell icon) at the top of the portal and clicking **Go to resource**

- Alternatively, you can click **All Services** on the left then enter **sql databases** in the search field and click **SQL databases** from the list, then click on your SQL database name (**Salesdb**) to open the SQL database blade.

The screenshot shows the Azure portal interface for a SQL database named 'Salesdb' under the server 'azserv'. The main content area displays the database's configuration, including its resource group (AzureDB-RG-16776558), status (Online), location (Brazil South), and subscription information. It also shows compute utilization over the last hour, which is currently at 3%. On the right, there is a circular progress bar indicating '0.2% USED SPACE' and detailed storage statistics: Used space (4 MB), Allocated space (16 MB), and Maximum storage size (2 GB). The left sidebar contains navigation links for Overview, Activity log, Tags, Diagnose and solve problems, Quick start, Query editor (preview), Power Platform (Power BI, Power Apps, Power Automate), Settings (Compute + storage, Connection strings, Properties, Locks), and Data management (Geo-Replication, Sync to other databases).

## Congratulations!

You have successfully completed this lab. Click **Next** to advance to the next lab. To save the lab, in the lab environment, on the **Exit** menu, select **Save**. Make sure to select the **Yes, save my lab** button as future labs are dependent on your work performed here.



# Various Tools to Manage Azure SQL Database

---

## Introduction

In this lab, you will learn how to connect to an Azure SQL Database by using the SQL Server Management Studio, SQL Server Data Tools and the Azure Portal.

## Estimated Time

30 minutes

## Objectives

After completing this lab, you will be able to:

- Connect to your database using the SQL Server Management Studio
- Connect to your database using the SQL Server Data Tools
- Connect to your database using the Azure Data Studio
- Connect to your database using the Azure Portal

## Logon Information

1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `T` `@lab.VirtualMachine(AzureSQLDB).Username` using `T` `@lab.VirtualMachine(AzureSQLDB).Password` as the password.

Click the `T` **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Connect to your logical server with SQL Server Management Studio

---

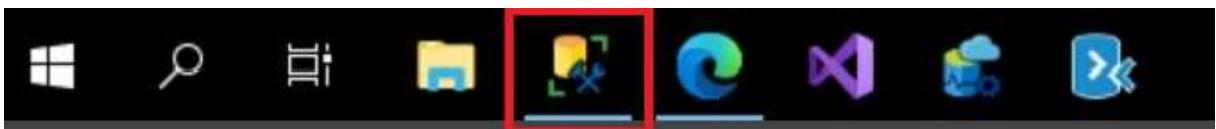
This exercise is to practice connecting to your Azure SQL Database with SQL Server Management Studio.

---

## Tasks

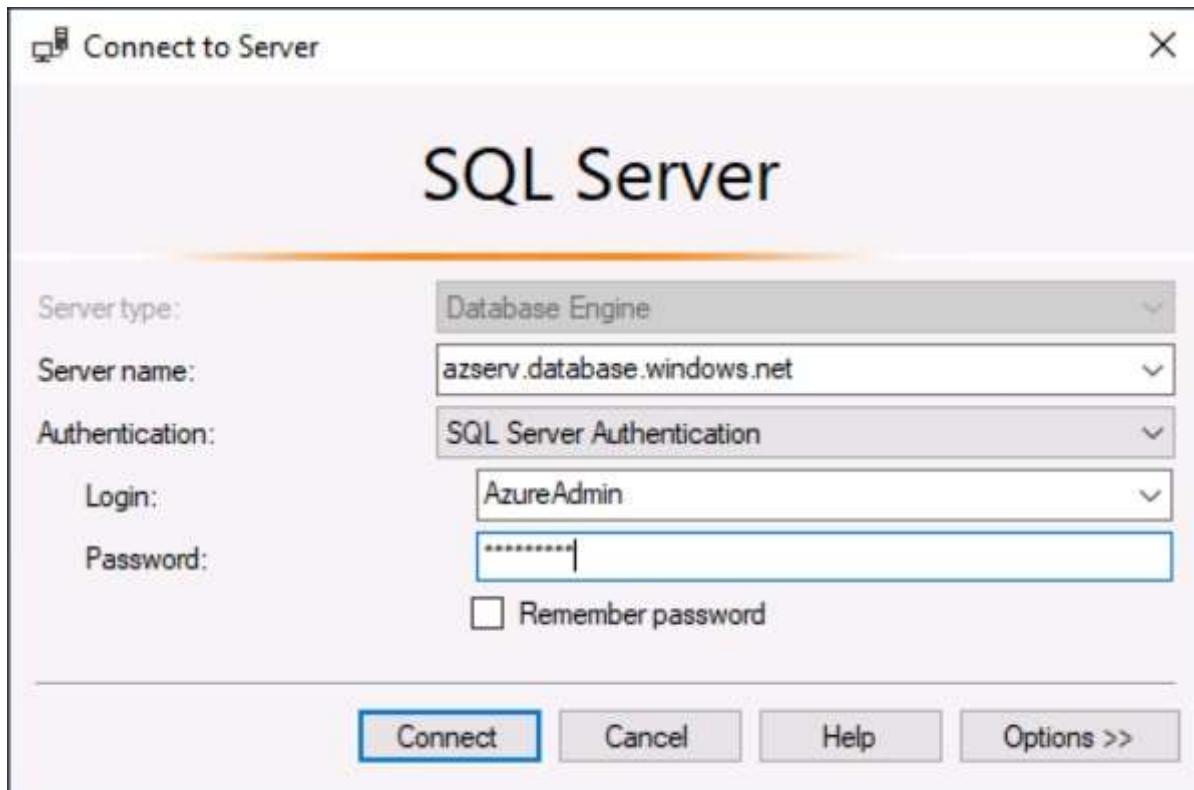
- 1. Connect to Azure with SSMS

Open **SQL Server Management Studio** (SSMS) by clicking the icon in the Windows Task Bar.



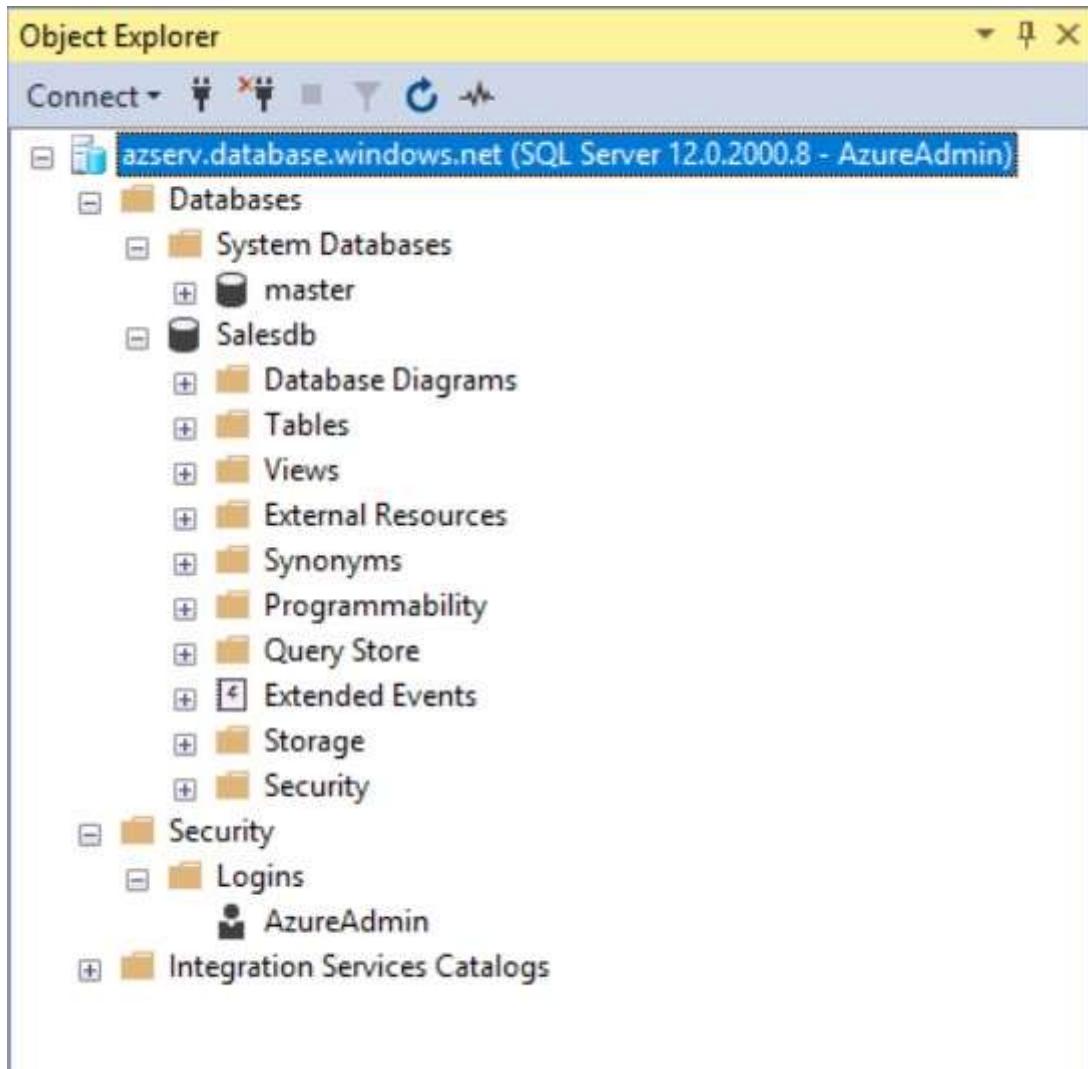
- 2. Enter the following information to connect to Azure logical server, and then click **Connect**.

Setting	Value
<b>Server name</b>	<input type="text"/> <LogicalServerName>.database.windows.net
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text"/> AzureAdmin
<b>Password</b>	<input type="text"/> Passw0rd1



3. Examine the Object Explorer

Examine the object explorer as it looks slightly different than a regular SQL Instance



- 4. Check the properties of the database

Right click **Salesdb**, select **Properties**

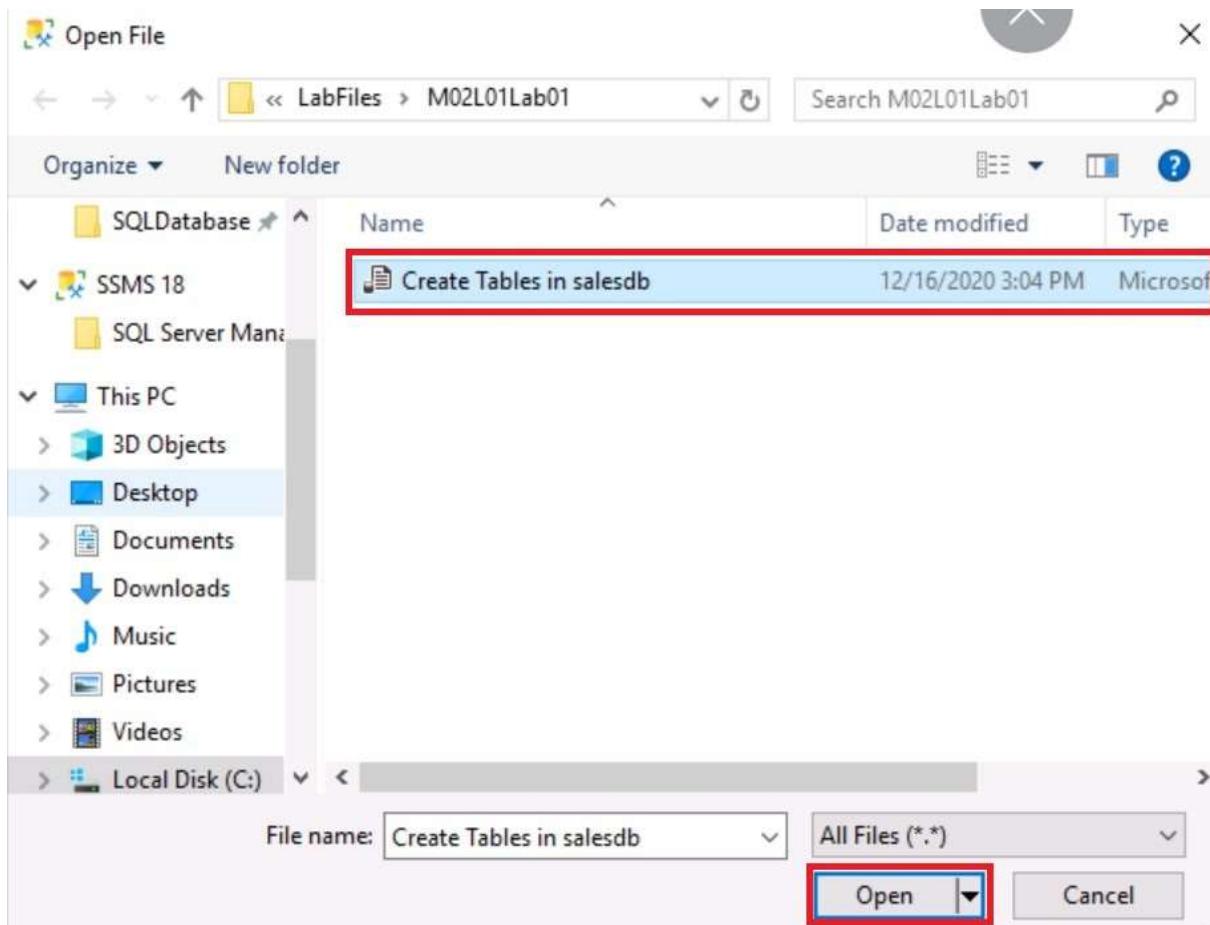
- 5. Go through the pages

Examine the different pages "General, Options, Configure SLO, Change Tracking, Permissions, Extended Properties, Query Store" and close the **Properties** window

- 6. Open a script to create the **dbo.Customers** table.

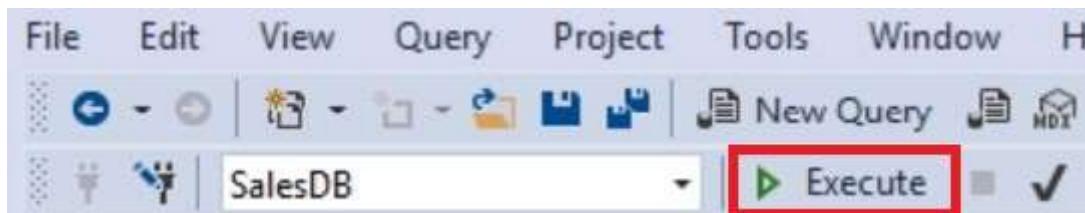
Navigate to **C:\LabFiles\M02L01Lab01** by using the **File/Open/File** menu option or the **Open File** icon on the SSMS Toolbar.

- 7. Select the **Create Tables in salesdb.sql** script and click the **Open** button.



- 8. Execute the script by clicking the **Execute** button.

**i** The script can take up to 1 minute to be executed



- 9. Click on the **New Query** button (right above the **Execute** button) and type the following code:

▶ `SQL SELECT * FROM dbo.Customers;`

- 10. Execute the statement by clicking the **Execute** button.

[!NOTE]The statement should return 1,000 rows.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

Thank you!

## Exercise 2: Connect to your logical server with SQL Server Data Tools

---

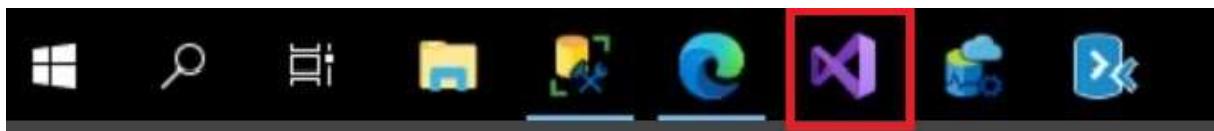
This exercise is to practice connecting to your Azure SQL Database with SQL Server Data Tools

---

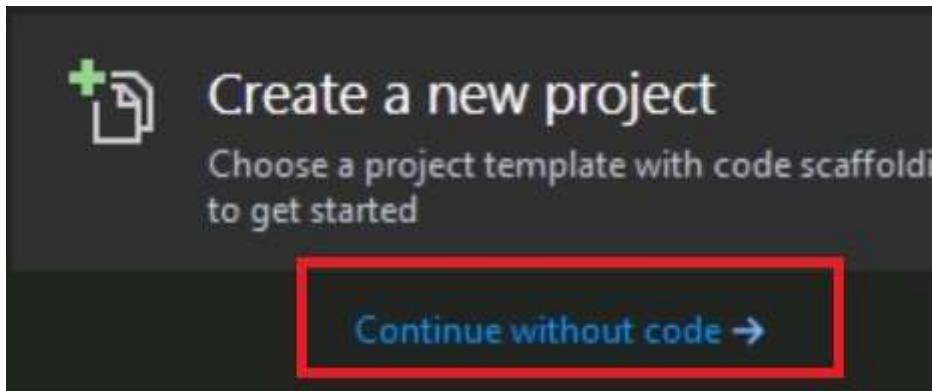
### Tasks

- 1. Open SSDT

Open **SQL Server Data Tools (SSDT)** by clicking the **Visual Studio** icon in the Windows Task Bar.

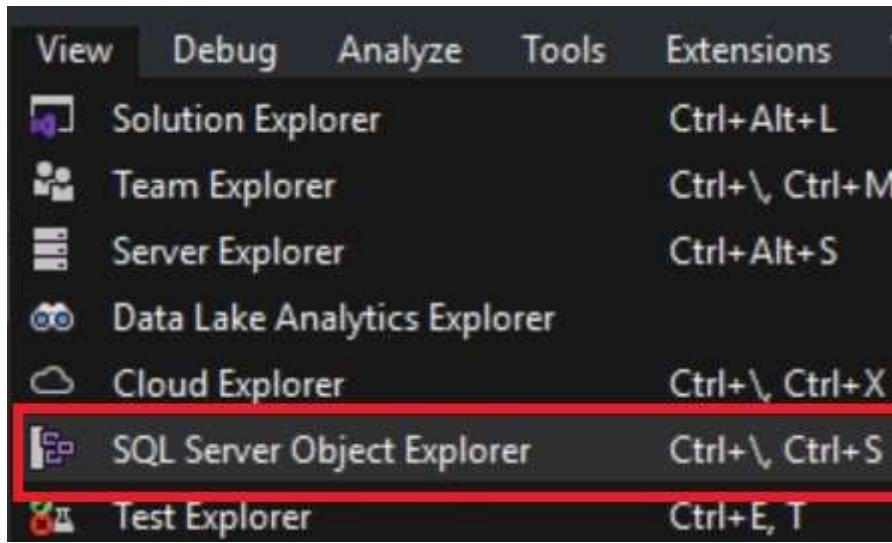


- 2. On the splash screen, Click on **Continue without code ->**



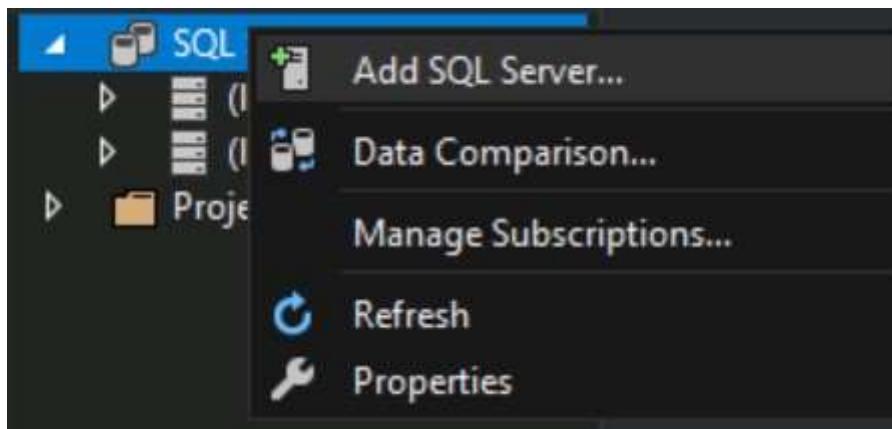
- 3. Open SQL Server Object Explorer

In the top menu select **View** and select **SQL Server Object Explorer**. This menu is going to appear on the left. If you cannot see it, try to close any other menu that it is open on the left first.



4. Add SQL Server

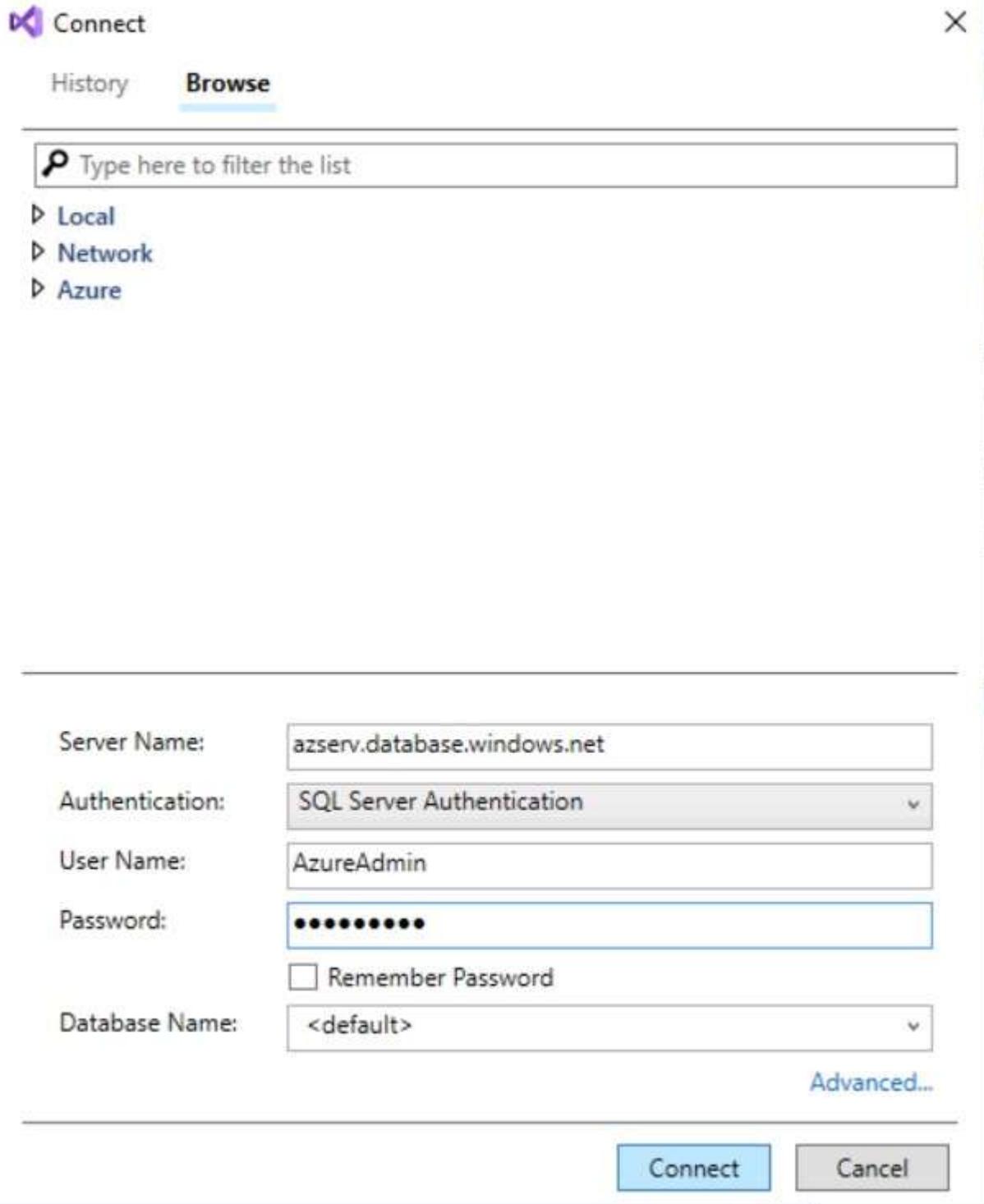
Right click **SQL Server** and select **Add SQL Server**



5. Connect to Azure with SSDT

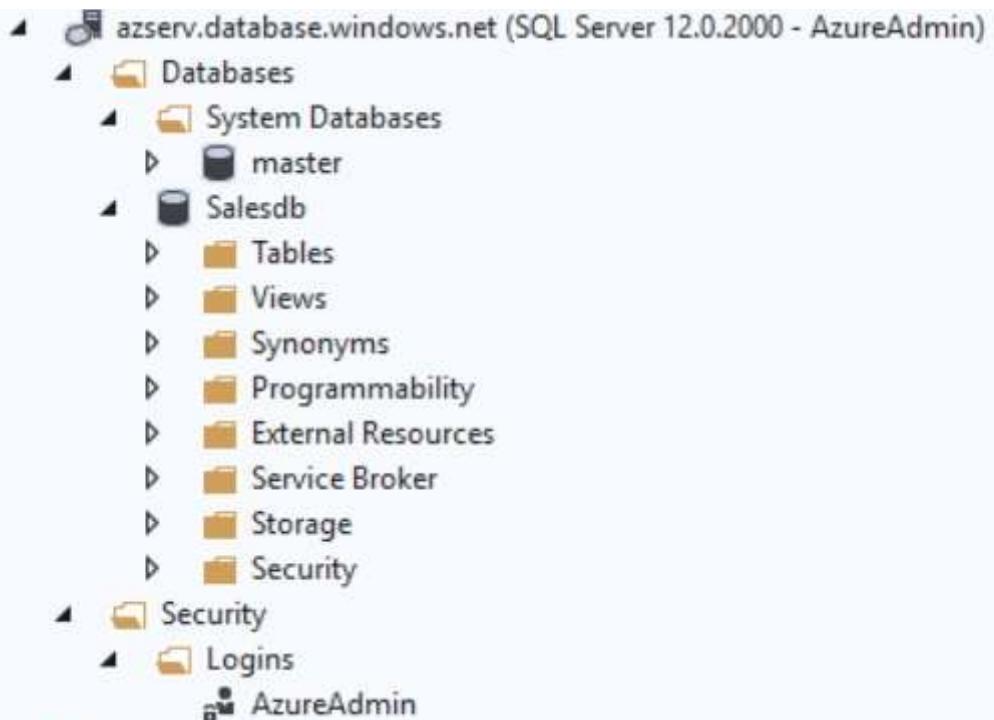
Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
Server name	<input type="text" value="&lt;LogicalServerName&gt;.database.windows.net"/>
Authentication	SQL Server Authentication
Login	<input type="text" value="AzureAdmin"/>
Password	<input type="text" value="Passw0rd1"/>
Database name	Default



6. Examine the Object Explorer

Examine the object explorer as it looks slightly different than a regular SQL Instance



Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

# Exercise 3: Connect to your logical server with Azure Data Studio

---

This exercise is to practice connecting to your Azure SQL Database with Azure Data Studio

---

## Tasks

- 1. Open Azure Data Studio

Open **Azure Data Studio** by clicking the icon in the Windows Task Bar.



- 2. Create a Connection

On the main screen, click on **Create a Connection**

- 3. Connect to Azure with Azure Data Studio

Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
<b>Server name</b>	<input type="text" value="T &lt;LogicalServerName&gt;.database.windows.net"/>
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text" value="T AzureAdmin"/>
<b>Password</b>	<input type="text" value="T Passw0rd1"/>
<b>Database name</b>	Default
<b>Server group</b>	Default

## Connection

### Recent Connections Saved Connections

 Clear List

CONTOSO, <default> (Windows Authentication)

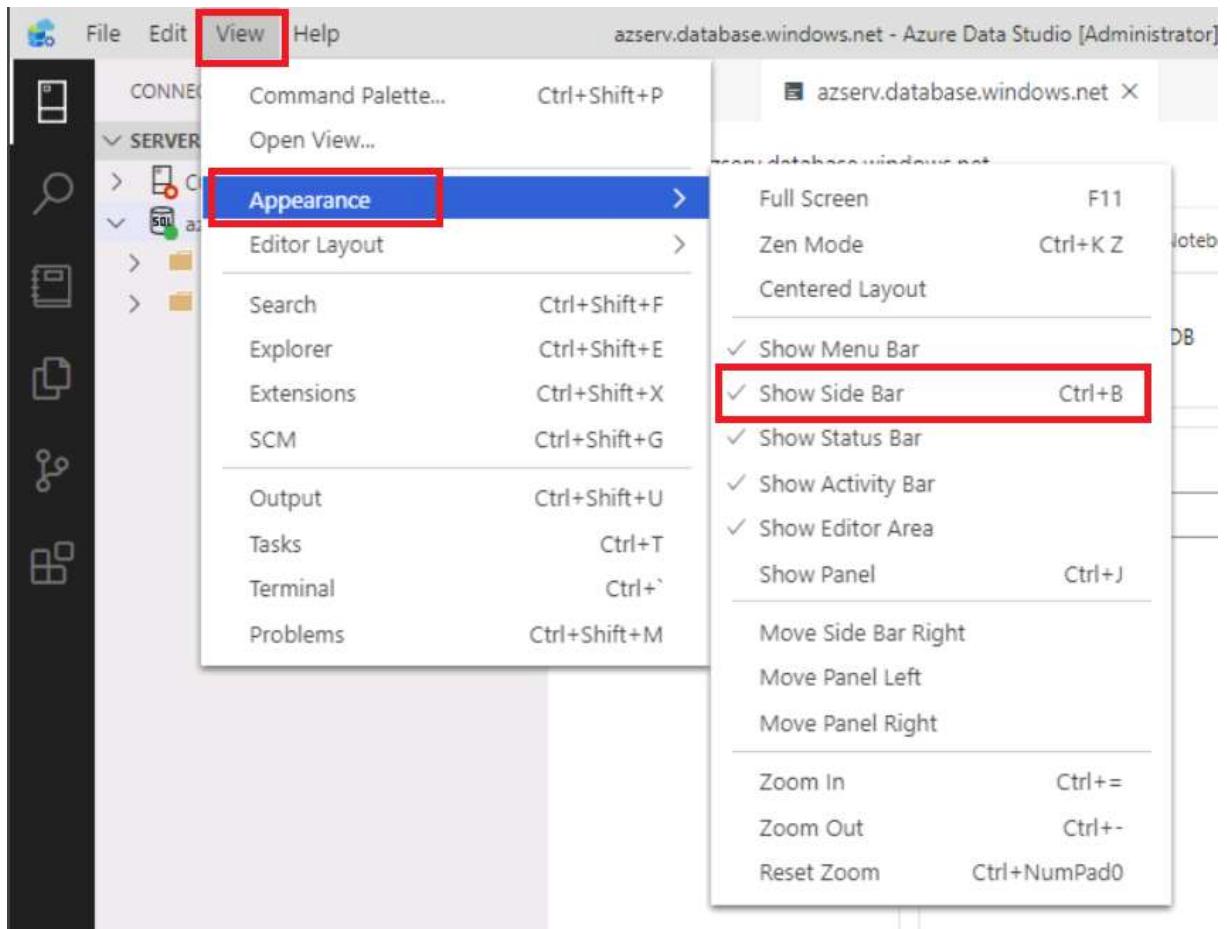
### Connection Details

Connection type	<input type="text" value="Microsoft SQL Server"/>
Server	<input type="text" value="azserv.database.windows.net"/>
Authentication type	<input type="text" value="SQL Login"/>
User name	<input type="text" value="AzureAdmin"/>
Password	<input type="password" value="*****"/>
<input type="checkbox"/> Remember password	
Database	<input type="text" value="&lt;Default&gt;"/>
Server group	<input type="text" value="&lt;Default&gt;"/>
Name (optional)	<input type="text"/>
<a href="#">Advanced...</a>	

[Connect](#)

[Cancel](#)

4. Go to the **View** menu on the top, Appearance and then Show Side Bar.



## □ 5. Examine the Object Explorer

Examine the object explorer and explore the environment

Azure Data Studio interface showing the Object Explorer on the left pane. The 'Servers' node is expanded, showing 'azserv.database.windows.net' and its 'Databases' node. The 'System Databases' node under 'Databases' is selected and highlighted with a red box. The main pane displays the details for the selected database, including version 12.0.2000.8 and type Azure SQL DB. A search bar at the bottom of the main pane shows results for 'master' and 'Salesedb'.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 4: Connect to your Azure SQL Database with Azure Portal

---

This exercise is to practice connecting to your Azure SQL Database with the Azure Portal.

---

### Tasks

- 1. Connect to Microsoft Azure Portal

Open web browser and navigate to <http://portal.azure.com> to connect to Microsoft Azure Portal. Login as `[T] @lab.CloudPortalCredential(LabUser).Username` using `[T] @lab.CloudPortalCredential(LabUser).Password` as the password.

- 2. Click SQL Database

In the left menu, click **SQL databases**

- 3. Click Salesdb Database

Click **Salesdb database**

- 4. Examine the options

Observe the Overview, Power Platform, Settings, Data Management, Integrations, Security, Intelligent Performance, Monitoring, Automation, Support + Troubleshooting sections under the SQL database blade. Close all open windows when finished.

Congratulations!

You have successfully completed this lab. Click **Next** to advance to the next lab.

# Scale up or down an Azure SQL Database

---

## Introduction

In this lab, you will learn how to scale up an Azure SQL Database and verify that the database has moved to the selected pricing tier.

## Estimated Time

10 minutes

## Objectives

At the end of this lab, you will be able to:

- Scale up an Azure SQL Database
- Verify that the database has moved to the selected pricing

## Logon Information

1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB).Username` using  `@lab.VirtualMachine(AzureSQLDB).Password` as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Scale up an Azure SQL Database

---

In this exercise, you're going to Scale up an Azure SQL Database.

---

## Tasks

- 1. Connect to Microsoft Azure Portal

Open web browser and navigate to <https://portal.azure.com> to connect to Microsoft Azure Portal. Login as `@lab.CloudPortalCredential(LabUser).Username` using `@lab.CloudPortalCredential(LabUser).Password` as the password.

- 2. Open SQL databases

Click **SQL databases** on the left side

- 3. Click the database

Click the database you want to change

- 4. Select Pricing Tier

Click **Compute + Storage** under the settings

Search (Ctrl+ /)

- Overview
- Activity log
- Tags
- Diagnose and solve problems
- Quick start
- Query editor (preview)

Power Platform

- Power BI (preview)
- Power Apps (preview)
- Power Automate (preview)

Settings

- Compute + storage
- Connection strings
- Properties
- Locks

□ 5. Change the Tier

Select a new tier, by clicking the **Standard** tile, and increase the performance by moving the DTUs slider to 20 (S1) and click **Apply**

Service tier: Standard (For workloads with typical performance requirements)

DTUs: What is a DTU?

Data max size (GB):

Backup storage redundancy:

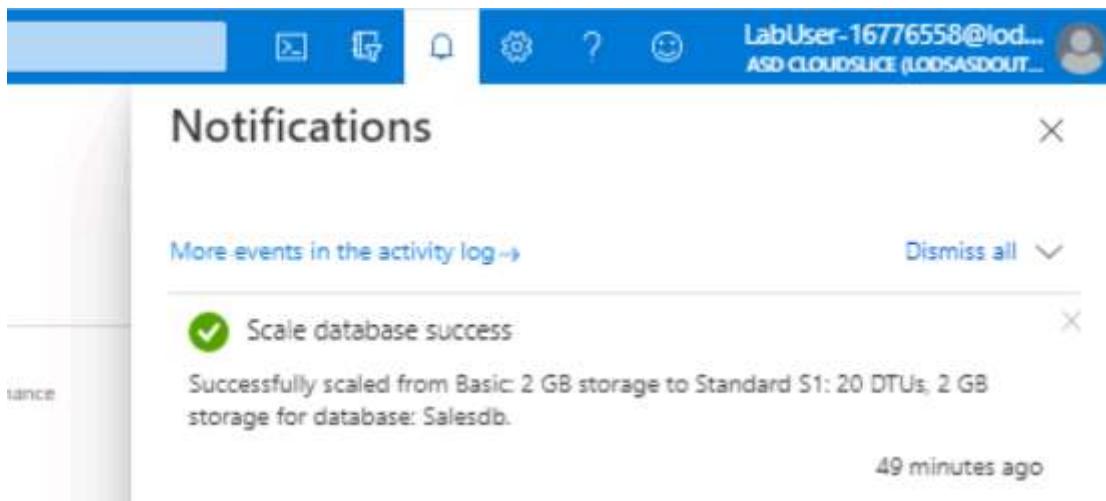
- Locally-redundant backup storage - Previous
- Zone-redundant backup storage - Previous
- Geo-redundant backup storage

⚠ Selected value for backup storage redundancy is Geo-redundant backup storage. Note that database backups will be geo-replicated to the paired region. Learn more ↗

**Apply**

6. Click the notification

- i** Clicking **Apply** submits a scale request to change the database tier. Depending on the size of your database the scale operation can take some time to complete. Click the Notifications icon for details and status of the scale operation.



Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Verify the database is at the selected pricing tier

---

After the scaling operation is complete inspect and confirm the database is at the desired tier.

---

### Tasks

- 1. Open SQL databases

Click **SQL databases** on the left side

- 2. Click the database

Click the database you updated

- 3. Click the Overview

Check the **Pricing tier** and confirm it is set to the correct tier (S1: 20 DTUs)

---

The screenshot shows the Azure portal interface for a database named 'Salesdb (azserv/Salesdb)'. The left sidebar has a 'Overview' button highlighted with a red box. The main content area displays various database settings under the 'Essentials' section. A second red box highlights the 'Pricing tier' field, which shows 'Standard S1: 20 DTUs'.

Setting	Value
Resource group (change)	AzureDB-RG-1d6776558
Status	Online
Location	Brazil South
Subscription (change)	ASD Developer 1
Subscription ID	cd5624ee-c42c-4f43-9c6d-4aea23072cf3
Pricing tier	Standard S1: 20 DTUs
Tags (change)	Click here to add tags

Congratulations!

You have successfully completed this lab. Click **Next** to advance to the next lab.

# Elastic Database Jobs

---

## Introduction

In this lab, you will learn how to create and schedule Elastic Database Jobs

## Estimated Time

30 minutes

## Objectives

At the end of this lab, you will be able to:

- Create Elastic Database Jobs
- Schedule Elastic Database Jobs

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as  **@lab.VirtualMachine(AzureSQLDB).Username** using  **@lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

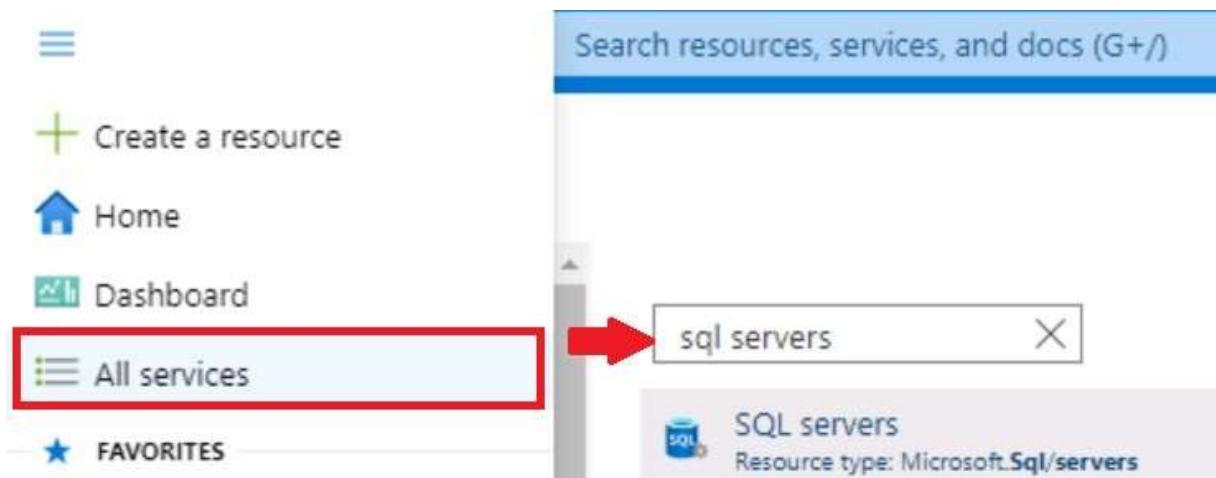
# Exercise 1: Create Elastic Job database and Elastic Job Agent

---

This exercise shows how to create Elastic Job database and Elastic Job Agent

## Tasks

- 1. Connect to the Azure Portal and browse to your SQL Server (logical server)
  - a. Open the Azure Portal in your browser [T] <https://portal.azure.com>
  - b. Login as [T] @lab.CloudPortalCredential(LabUser).Username using [T] @lab.CloudPortalCredential(LabUser).Password as the password.
  - c. Click **All Services** on the left then enter [T] sql servers in the search field and click **SQL servers** from the list.



- 2. Click on your SQL server name to open the SQL server blade.
- 3. To add Firewall rules to the server, under the **Security** blade, click **Networking** to show firewall settings.

## Security

### Networking

Microsoft Defender for Cloud

Transparent data encryption

Identity

Auditing

- 4. Click **Add a firewall rule**: Name the rule name as **Elastic Jobs** and provide the Start IP of **0.0.0.0** and the End IP **255.255.255.255**.

#### Firewall rules

Allow certain public internet IP addresses to access your resource. [Learn more](#)

+ Add your client IPv4 address

+ Add a firewall rule

Add a firewall rule

Rule name

Elastic Jobs

Start IP

0.0.0.0

End IP

255.255.255.255

OK

Cancel

- 5. Click **OK** and then **Save**.

**i** Best practice would be to use an IP address range from your Azure network.

- 6. Create 2 new empty databases

[!TIP] Review **Exercise 3** from the **Create an Azure SQL Database** lab, on how to create new databases (**Module 1 - Lesson 2 - Lab 1**).

**i** Make sure the databases have at least a S0 or higher service tier (Standard - 10 DTUs).

- a. Create a new empty database named **T JobDatabase**. This will be used as the Job database during Elastic Job Agent creation.
- b. Create a new empty database **T PerfResults**.

**i** Please wait until both databases are created before creating an Elastic Job Agent.

- 1. Create an Elastic Job Agent
  - a. In the top search bar type **T Elastic Job Agents** and hit **Enter**
  - b. Click on the **Create elastic job agent** button or **+ Create**
  - c. Configure the Elastic Job Agent with the following settings

Setting	Value
<b>Name</b>	<b>T MyJobAgent</b>
<b>Subscription</b>	Leave default
<b>Job Database</b>	Click <b>Select Job database</b> . Select your logical server <b>&lt;LogicalServerName&gt;</b> . Select the <b>JobDatabase</b> created earlier

- a. Click **OK** and then click **Create**. Wait for deployment to complete.

Congratulations!

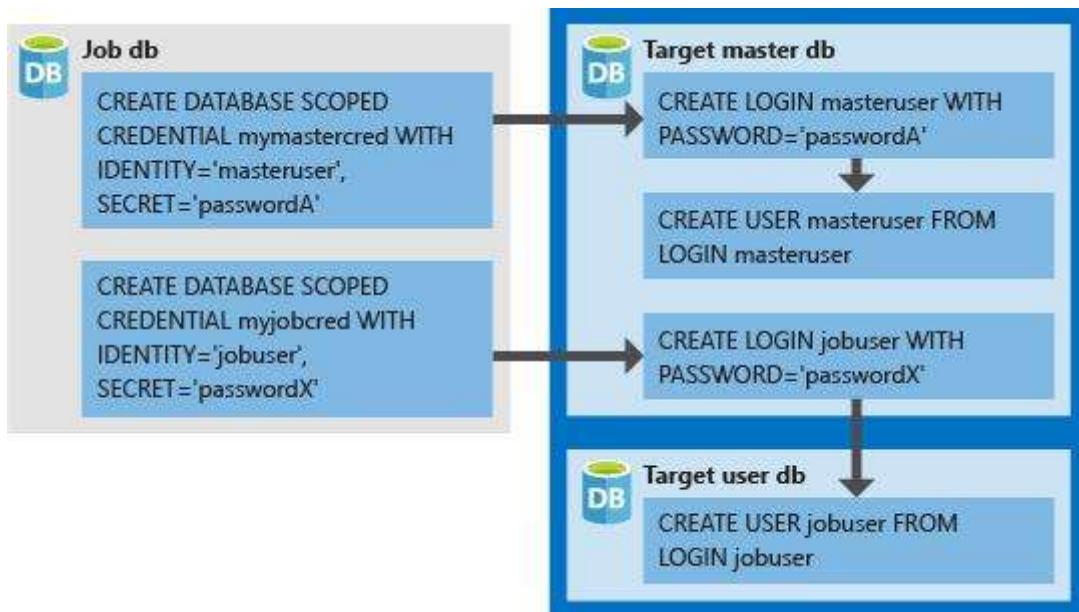
You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Create an Elastic Database Job

This exercise shows how to create an Elastic Database Job

### Tasks

The following image is designed to assist in understanding and setting up the proper job credentials. Remember to create the user in every database (all target user dbs) the job needs to run.



1. Create a credential for job execution. Connect to your **JobDatabase**.

Use SQL Server Management Studio (SSMS) to connect to the **Job Database**. Review the **Using Various Tools** lab, Exercise 1 as a reminder on connecting to a database. You can use the **Available Databases** drop-down in the top left of SSMS on the Toolbar.

- a. Create a database master key if one does not already exist, using your own password. You can use the statement below:

SQL

```
▶ CREATE MASTER KEY ENCRYPTION BY PASSWORD='ReallyD1ff1cultP@ssw0rd!';
```

- b. Create a database scoped credential. You can use the statement below:

SQL

```
▶ CREATE DATABASE SCOPED CREDENTIAL myjobcred WITH IDENTITY = 'jobcred'
```

```
SECRET = 'ReallyD1ff1cultP@ssw0rd!';
```

**i** Ignore any Intellisense errors

- c. Create a database scoped credential **for the master** database

- Make sure you are still connected to the **JobDatabase** database

SQL

```
► CREATE DATABASE SCOPED CREDENTIAL mymastercred WITH IDENTITY = 'master'  
SECRET = 'ReallyD1ff1cultP@ssw0rd!';
```

- 2. Create logins **in the master** database

Open a new query window and connect to the **master** database

- You can use the **Available Databases** drop-down in the top left of SSMS on the Toolbar.

Create the following logins:

SQL

```
► CREATE LOGIN mastercred WITH PASSWORD='ReallyD1ff1cultP@ssw0rd!'  
CREATE USER mastercred FROM LOGIN mastercred  
CREATE LOGIN jobcred WITH PASSWORD='ReallyD1ff1cultP@ssw0rd!'
```

- 3. Create a database user in **every database** where you want to execute the job

Open a new query window and connect to each user database by right-clicking the database and choosing **New Query**, then create the database user **jobcred** using the following code:

SQL

```
► CREATE USER jobcred FROM LOGIN jobcred  
ALTER ROLE db_owner ADD MEMBER jobcred
```

- 4. Configure a Target Group

**i** The job is going to be executed against all the databases in a logical server. We need to create a target group and add a server member to the group.

Open a new query window and connect to the **JobDatabase** database

Execute the following command to add a new Target Group

SQL

▶ `EXEC jobs.sp_add_target_group 'ServerGroup1'`

Execute the following command to add a server target member to the target group.

- ℹ Make sure the **server\_name** variable contains your logical server name before you execute the script.

SQL

▶ `EXEC jobs.sp_add_target_group_member 'ServerGroup1',  
@target_type = 'SqlServer',  
@refresh credential_name='mymastercred', --credential required to refresh the target  
@server_name=<LogicalServerName>.database.windows.net'`

## □ 5. Add a job

Open a new query window and connect to the **JobDatabase** database

Execute the following statement to add a new job that will be scheduled every minute to collect the performance data of all the databases

SQL

▶ `EXEC jobs.sp_add_job @job_name ='ResultsJob', @description='Collection Performance Data from All Databases',  
@schedule_interval_type='Minutes',  
@schedule_interval_count=15`

## □ 6. Add a job step

Open a new query window and connect to the **JobDatabase** database

Execute the following statement to add a new job step. The job step will capture all the performance data of every database and store it into a table.

- ℹ Make sure the **server\_name** variable contains your logical server name before you execute the script.

- ℹ There is no need to create the ResourceStats table in advance. If the table doesn't exist, it will be created automatically.

SQL

▶ `EXEC jobs.sp_add_jobstep  
@job_name='ResultsJob',`

```
@command= N'SELECT DB_NAME() DatabaseName, $(job_execution_id) AS job_exec
FROM sys.dm_db_resource_stats
WHERE end_time > DATEADD(mi, -20, GETDATE());',
@credential_name='myjobcred',
@target_group_name='ServerGroup1',
@output_type='SqlDatabase',
@output_credential_name='myjobcred',
@output_server_name='<LogicalServerName>.database.windows.net',
@output_database_name='PerfResults',
@output_table_name='ResourceStats'
```

## □ 7. Execute the job

Open a new query window and connect to the **JobDatabase** database

Execute the following statements to start the job manually

SQL

▶ EXEC jobs.sp\_start\_job 'ResultsJob'

## □ 8. Monitor job execution status

Open a new query window and connect to the **JobDatabase** database

Execute the following statements to view the execution status for all jobs

SQL

▶ --Connect to the job database specified when creating the job agent  
--View top-level execution status for the job named ◆ResultsJob◆  
SELECT \* FROM jobs.job\_executions  
WHERE job\_name = 'ResultsJob' and step\_id IS NULL  
ORDER BY start\_time DESC  
  
--View all top-level execution status for all jobs  
SELECT \* FROM jobs.job\_executions WHERE step\_id IS NULL  
ORDER BY start\_time DESC  
  
--View all execution statuses for job named ◆ResultsJob◆  
SELECT \* FROM jobs.job\_executions  
WHERE job\_name = 'ResultsJob'  
ORDER BY start\_time DESC  
  
-- View all active executions  
SELECT \* FROM jobs.job\_executions  
WHERE is\_active = 1  
ORDER BY start\_time DESC

**i** The all above queries are available in a file  
**C:\LabFiles\M02L03Lab01\16918\_M02\_L03\_Lab.sql.**

- 9. Close all open windows.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Point in time restore of an Azure SQL Database

---

## Introduction

In this lab, you will learn how to perform a point in time restore over the original database.

## Estimated Time

20 minutes

## Objectives

At the end of this lab, you will be able to:

- Perform a point in time restore over the original database

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as **T** **@lab.VirtualMachine(AzureSQLDB).Username** using **T** **@lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the **T Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Perform a point in time restore over the original database

---

This exercise shows the point in time restore of the database as a different database.

---

## Tasks

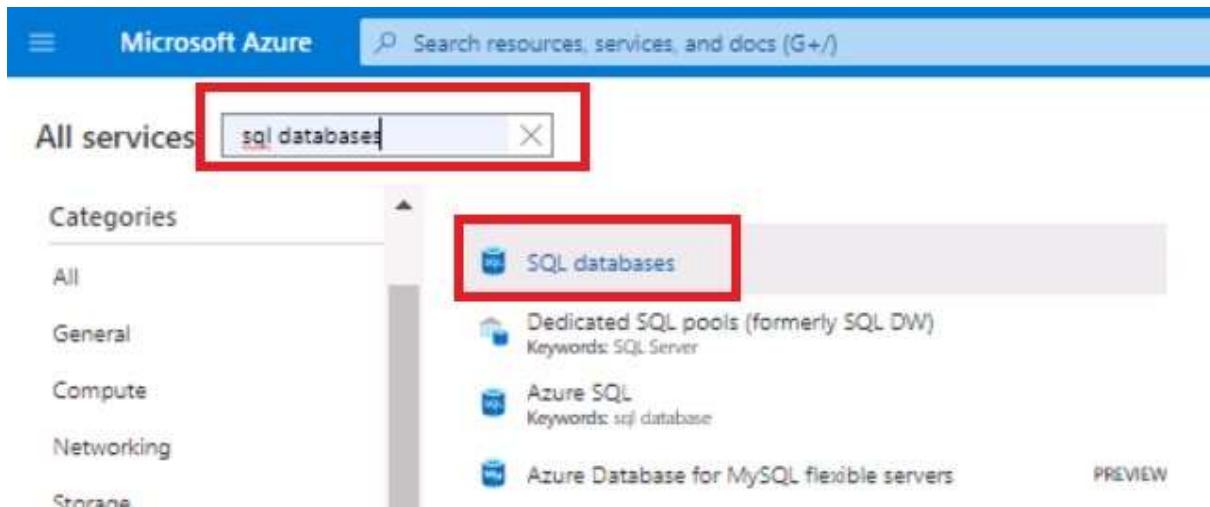
- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `T @lab.VirtualMachine(AzureSQLDB).Username` using `T @lab.VirtualMachine(AzureSQLDB).Password` as the password.

- 2. Connect to Microsoft Azure Portal

Open web browser and navigate to `T https://portal.azure.com` to connect to Microsoft Azure Portal. Login as `T @lab.CloudPortalCredential(LabUser).Username` using `T @lab.CloudPortalCredential(LabUser).Password` as the password.

- 3. Navigate to **Salesdb** database

Click **All Services**, fill in `T SQL databases` and Click **SQL Databases**



- 4. Open **Salesdb** database

Click **Salesdb** to open the SQL Database

All services >

## SQL databases

ASD Cloudslice (lodsasdoutlook.onmicrosoft.com)

[+ Add](#) [Reservations](#) [Manage view](#) [Refresh](#)

Filter for any field...

Subscription == all

Resource

Showing 1 to 3 of 3 records.

Name ↑

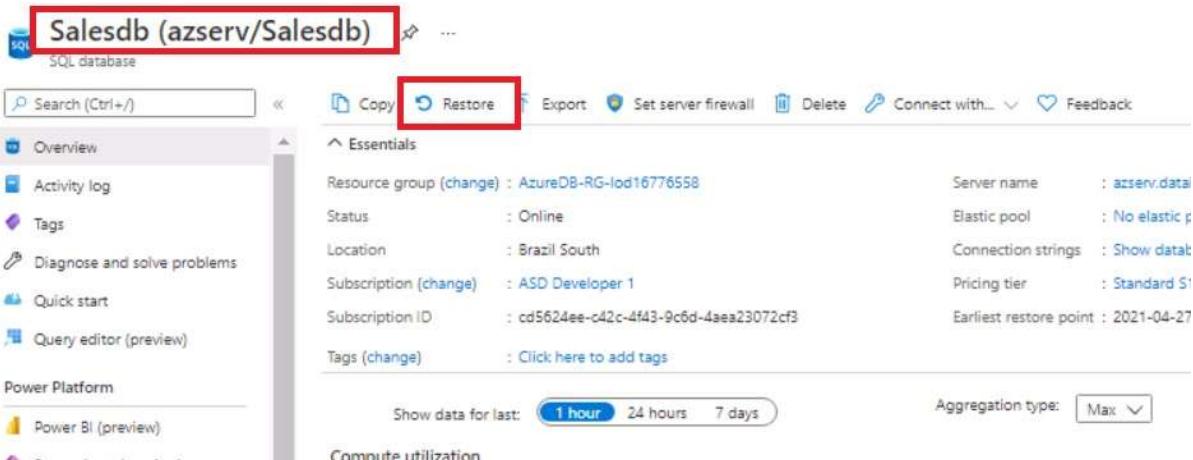
 JobDatabase (azserv/JobDatabase)

 PerfResults (azserv/PerfResults)

 Salesdb (azserv/Salesdb)

### □ 5. Click **Restore**

Click **Restore** to see the point-in-time restore option



Salesdb (azserv/Salesdb)   ...

SQL database

Search (Ctrl+ /)   Export Set server firewall Delete Connect with... Feedback

Overview  Activity log  Tags  Diagnose and solve problems  Quick start  Query editor (preview) 

Power Platform  Power BI (preview) 

Essentials

Resource group (change)	: AzureDB-RG-lod16776558	Server name	: azserv.datal
Status	: Online	Elastic pool	: No elastic p
Location	: Brazil South	Connection strings	: Show data
Subscription (change)	: ASD Developer 1	Pricing tier	: Standard S1
Subscription ID	: cd5624ee-c42c-4f43-9c6d-4aea23072cf3	Earliest restore point	: 2021-04-27
Tags (change)	: Click here to add tags		

Show data for last:    Aggregation type:

Compute utilization

### □ 6. Configure Restore Options

On the Restore page, select the date and time (in UTC time) to restore the database to, click **Review + create**.

- In **Restore Point (UTC)**, choose a restore point between earliest restore point and latest backup time

## Create SQL Database - Restore database

Microsoft

Source Database:

Salesdb

Select source

Point-in-time

Earliest restore point

2021-04-27 19:20 UTC

Restore point (UTC) \*

04/29/2021

2:32:00 PM

i Choose a restore point between the earliest restore point and the current time in UTC.

### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

Salesdb\_2021-04-29T14-32Z

Server \*

azserv (Brazil South)

Want to use SQL elastic pool? \*

Yes  No

Compute + storage \*

Standard S1

20 DTUs, 2 GB storage

[Configure database](#)

### Backup storage redundancy

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy

- Locally-redundant backup storage - Preview
- Zone-redundant backup storage - Preview
- Geo-redundant backup storage

i If no option is selected, backup storage redundancy will be set by default based on the redundancy setting of the source database.

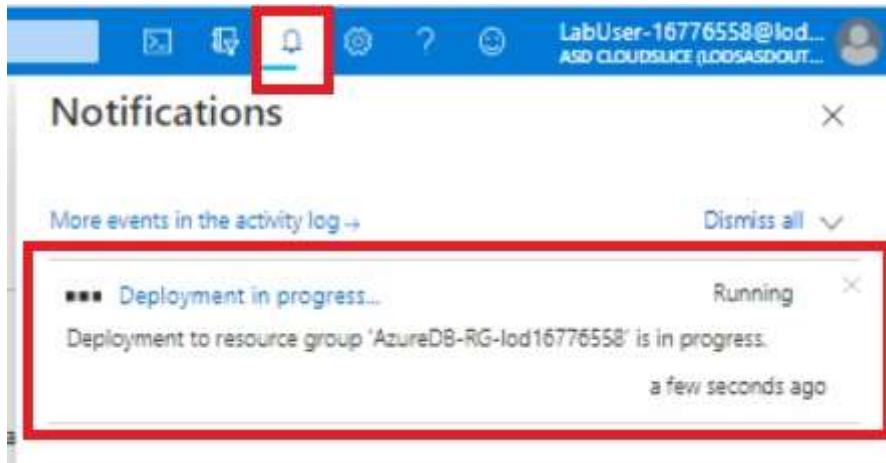
**Review + create**

[Next : Review + create >](#)

7. Click **Create**

8. Check the notifications

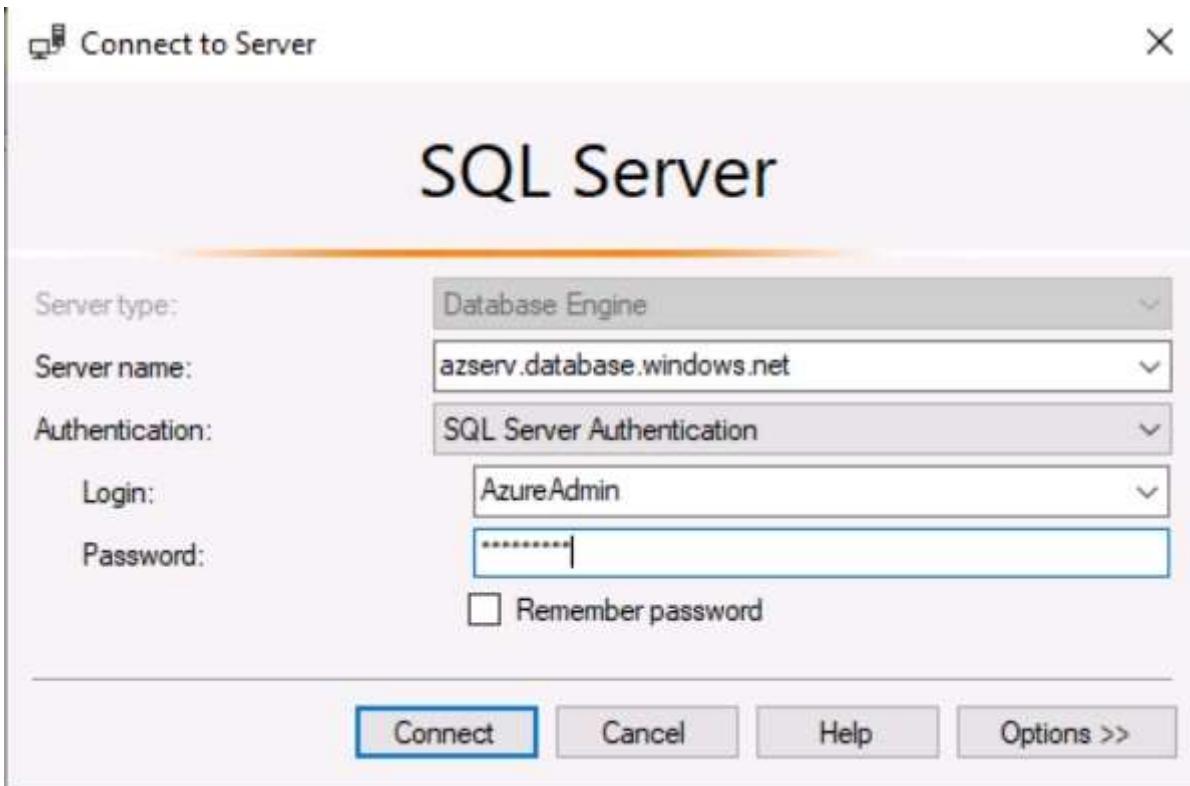
Click the notification icon at the top upper right of the page. Wait for the restore operation to complete.



9. Connect to Logical Server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the required information to connect to Azure logical server. Click **Connect**

Setting	Value
<b>Server Name</b>	<input type="text" value="T &lt;LogicalServerName&gt;.database.windows.net"/>
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text" value="T AzureAdmin"/>
<b>Password</b>	<input type="text" value="T Passw0rd1"/>



- 10. Check the restore status with a query

Open a new query window in SSMS and run the query below:

- Make sure you are connected to the **master** database using the Available Databases dropdown in the top left of the SSMS Toolbar.

SQL

```
> select * from sys.dm_operation_status
```

select \* from sys.dm\_operation\_status

session_activity_id	resource_type	resource_type_desc	major_resource_id	minor_resource_id	operation	state	state_desc	percent_complete
4847A5D9-A907-4628-8E5E-E664E74FE857	0	Database	salesdb		CREATE DATABASE	2	COMPLETED	100
B3400EB4-637D-4896-B832-DFC1F34FCC26	0	Database	salesdb_2017-02-22T01-27Z		DATABASE RESTORE	2	COMPLETED	100

- 11. Return to the Azure Portal and Navigate to the SQL Databases

Click **All Services**, fill in **SQL databases** and click **SQL Databases**

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+/-)" and a red box highlighting the search term "sql databases". Below the search bar, the "All services" category is selected. On the left, a sidebar lists categories: All, General, Compute, Networking, and Storage. In the main pane, a red box highlights the "SQL databases" service, which is listed under the "All" category. Other options shown are "Dedicated SQL pools (formerly SQL DW)" and "Azure SQL". To the right, "Azure Database for MySQL flexible servers" is listed with a "PREVIEW" tag.

□ 12. Examine the SQL Databases in Portal

Check the recently restored database

The screenshot shows the "SQL databases" blade in the Azure portal. The title bar says "All services > SQL databases". The main area displays a table of databases:

Name	Server	Replica type	Pricing tier	Location
JobDatabase (azserv/JobDatabase)	azserv	--	Standard S0: 10 DT...	Brazil South
PerfResults (azserv/PerfResults)	azserv	--	Standard S0: 10 DT...	Brazil South
Salesdb (azserv/Salesdb)	azserv	--	Standard S1: 20 DT...	Brazil South
Salesdb_2021-04-29T14-32Z (azserv/Salesdb_2021-04-29T14-32Z)	azserv	--	Standard S1: 20 DT...	Brazil South

A red box highlights the last two rows of the table, which represent the recently restored databases.

□ 13. Examine the SQL Databases in SSMS

Check the recently restored database in SSMS

The screenshot shows the Object Explorer in SSMS. The connection path is "azserv.database.windows.net (SQL Server 12.0.2000.8 - AzureAdmin)". Under the "Databases" node, a red box highlights the following databases:

- System Databases
- JobDatabase
- PerfResults
- Salesdb
- Salesdb\_2021-04-29T14-32Z

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Rename old and new databases

---

This exercise shows how to rename old and new databases

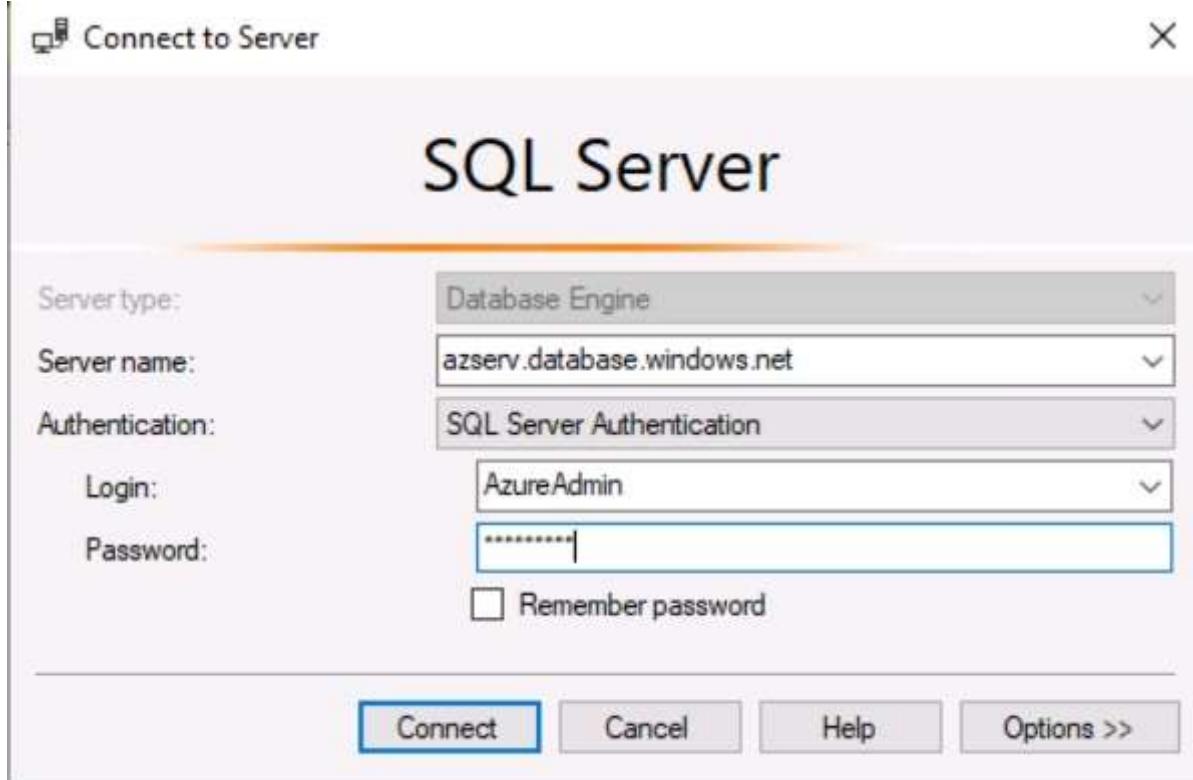
---

### Tasks

- 1. Connect to Logical Server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the required information to connect to Azure logical server. Click **Connect**.

Setting	Values
<b>Server Name</b>	<input type="text" value="T &lt;LogicalServerName&gt;.database.windows.net"/>
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text" value="T AzureAdmin"/>
<b>Password</b>	<input type="text" value="T Passw0rd1"/>



- 2. Rename the original database

Rename the original database to something else using the query on the server's **master** database:

SQL

▶ ALTER DATABASE [salesdb] MODIFY NAME = [salesdb\_old];

3. Rename the restored database

Rename the newly restored database to the original name using the query on the server's **master** database:

**i** Change the newly restored database name to the actual name for your restore operation.

SQL

▶ ALTER DATABASE [salesdb.2017-02-22T01-27Z] MODIFY NAME = [salesdb]

4. Close all open windows.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Configure Geo Replication for an Azure SQL Database

---

## Introduction

In this lab, you will learn how to configure Failover Groups and Geo Replication for an Azure SQL Database

## Estimated Time

20 minutes

## Objectives

After completing this lab, you will be able to:

- Configure Failover Groups
- Verify the functionality of the secondary replica
- Perform a failover to the secondary replica

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as  **@lab.VirtualMachine(AzureSQLDB).Username** using  **@lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Setup Failover Groups

---

This exercise shows how to setup Failover Groups

## Tasks

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `T @lab.VirtualMachine(AzureSQLDB).Username` using `T @lab.VirtualMachine(AzureSQLDB).Password` as the password.

- 2. Connect to Microsoft Azure Portal

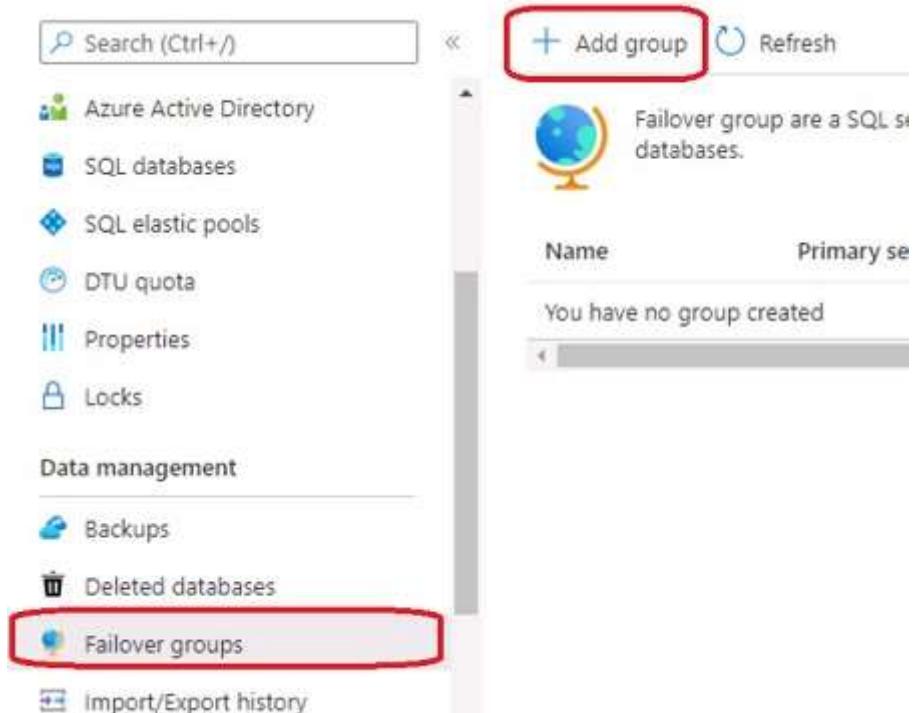
Open web browser and navigate to `T https://portal.azure.com` to connect to Microsoft Azure Portal. Login as `T @lab.CloudPortalCredential(LabUser).Username` using `T @lab.CloudPortalCredential(LabUser).Password` as the password.

- 3. Navigate to your logical server

Navigate to your logical server that you have created in Module 1.

- 4. Create Failover Group

Click on **Failover Groups** and click on the button **Add group**.



- 5. Specify the following settings for your Failover Group:

Setting	Value
<b>Failover Group Name</b>	Define a unique name
<b>Secondary Server</b>	Create a new server
<b>Server Name</b>	A unique server name
<b>Server Admin Login</b>	<input type="text" value="AzureAdmin"/>
<b>Password</b>	<input type="text" value="Passw0rd1"/>
<b>Confirm password</b>	<input type="text" value="Passw0rd1"/>
<b>Location</b>	Choose a location that is close to the location of your first logical server but different. Click <b>Select</b>
<b>Read/Write Failover Policy</b>	Automatic
<b>Read/Write Grace Period (hours)</b>	1 hour
<b>Databases within the group</b>	Click <b>Configure database</b> . Select the <b>salesdb</b> . Click <b>Select</b>

Click the **Create** button and wait until the failover group is deployed

Record the name of your failover group in the following text box:

#### **Failover group**

#### 6. Review the Replication

Once the Failover Group is created, click on the group to see the replication and configuration details.

azservfailover ...

azserv

Save  Discard  Add databases  Edit configuration  Remove databases  Failover  Forced Failover  Delete

Configuration details Databases within group Databases selected to be added (0) Databases selected for removal (0)

Server	Role	Read/Write failover policy	Grace period
azserv (Brazil South)	Primary	Automatic	1 hours
azservsec (Central US)	Secondary		

Read/write listener endpoint  
azservfailover.database.windows.net

Read-only listener endpoint  
azservfailover.secondary.database.windows.net

7. Add Firewall rules to your Secondary Server

Navigate to the secondary server that you have just created. Click **Firewall and virtual networks** to show firewall settings.

Click **Add client IP**. Start IP and End IP will be set automatically.

Rename the rule name as **T MS Office Target** and Click **Save**.

8. On the **Success!** dialog box click **OK**.

The screenshot shows the Azure portal interface for managing a database server. The left sidebar lists various management options like SQL elastic pools, DTU quota, Properties, Locks, Data management, Security, Intelligent Performance, and Automatic tuning. The 'Firewalls and virtual networks' option is selected and highlighted with a red box. At the top right, there are 'Save' and 'Discard' buttons, and a prominent 'Add client IP' button highlighted with a red box. The main pane displays settings for network access, including 'Deny public network access' (set to No), 'Minimum TLS Version' (set to 1.2), 'Connection Policy' (set to Default), and 'Allow Azure services and resources to access this server' (set to Yes). A table lists a single rule: 'MS Office Target' with Start IP 199.101.110.14 and End IP 199.101.110.14. Below this, sections for 'Virtual networks' and 'No vnet rules for this server.' are shown.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Verify the functionality of the secondary database

---

This exercise shows the functionality of the secondary database

---

### Tasks

- 1. Connect to the secondary server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
<b>Server name</b>	Use the Read-only listener endpoint that you can find on your Failover Group
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text" value="AzureAdmin"/>
<b>Password</b>	<input type="password" value="Passw0rd1"/>

- 2. Verify the database on the secondary server

You can verify that the database has been created on the secondary server

- 3. Connect to the primary server

Within **SSMS** Object Explorer, click the **Connect** drop-down and choose **Database Engine**. Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
<b>Server name</b>	Use the Read/write listener endpoint that you can find on your Failover Group
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text" value="AzureAdmin"/>
<b>Password</b>	<input type="password" value="Passw0rd1"/>

- 4. Create a table on the Primary

- You can test the data is being properly replicated across the secondary by creating a table on the primary and populate it.

- a. Open a new Query Window

Right click on the **Salesdb** on the primary server and select **New Query**.

- a. Execute the script

Open the script **C:\LabFiles\M03L02Lab01\16918\_M03\_L02\_Lab.sql** by using the **File/Open/File** menu option or the **Open File** icon on the SSMS Toolbar. After opening the script, click **Execute**.

- i** Verify that you are connected to the primary server and the Salesdb database by checking the yellow status bar at the bottom of the query tab.

- 5. Query the table on the secondary server

Within SSMS Object Explorer under the secondary server, right-click on the **Salesdb**, choose **New Query** then run the query below. The query should succeed to show the new table has been replicated to the secondary server.

- i** Verify that you are connected to the secondary server and the Salesdb database by checking the yellow status bar at the bottom of the query tab.

SQL

▶ `SELECT * FROM Person`

- 6. Update the table on the secondary server

Connect to the secondary server. Open a new query window and run the query below.

- It fails because database is in a readable state.

SQL

▶ `UPDATE [Person] SET FirstName = 'Jack' WHERE [BusinessEntityID] = 1`

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 3: Perform a Failover

---

This exercise shows the failover

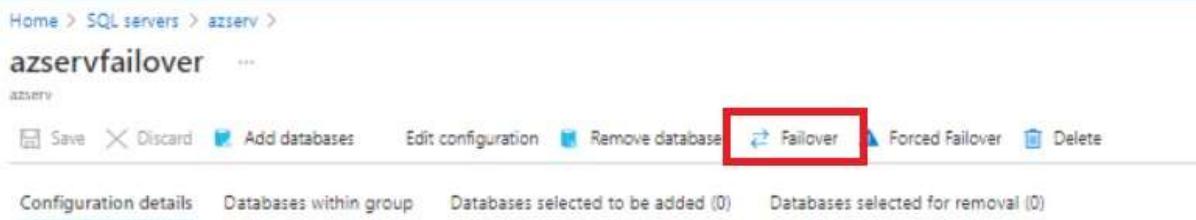
---

### Tasks

- 1. Perform a failover

    Navigate to the **T <FailoverGroup>** Failover Group that you have created in Exercise 1.

    Click on it to show the configuration details. To perform a failover, click the **Failover** button, confirm by clicking **Yes** and wait until the failover is finished.



- 2. Connect to the Primary Logical Server

    Within **SSMS**, switch to the query tab that is connected with the following connection information:

Setting	Value
<b>Server name</b>	Use the Read/write listener endpoint that you can find on your Failover Group
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<b>T AzureAdmin</b>
<b>Password</b>	<b>T Passw0rd1</b>

**i** Verify that you are connected to the primary server and the Salesdb database by checking the yellow status bar at the bottom of the query tab.

- 3. Verify the new Primary

    Execute the query below to verify if the secondary became the new primary.

SQL

▶ SELECT @@SERVERNAME

- 4. Close all open windows without saving changes.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Implement Always Encrypted

---

## Introduction

In this lab, you will learn how to implement Always Encrypted

## Estimated Time

20 minutes

## Objectives

After completing this lab, you will be able to:

- Implement Always Encrypted
- Use .net app to test Always Encrypted

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as T  
**@lab.VirtualMachine(AzureSQLDB).Username** using T  
**@lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the T **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Implement Always Encrypted on Azure SQL Database

This exercise shows how to implement Always Encrypted.

## Tasks

- 1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as **T @lab.VirtualMachine(AzureSQLDB).Username** using **T @lab.VirtualMachine(AzureSQLDB).Password** as the password.

- 2. Connect to Microsoft Azure Portal

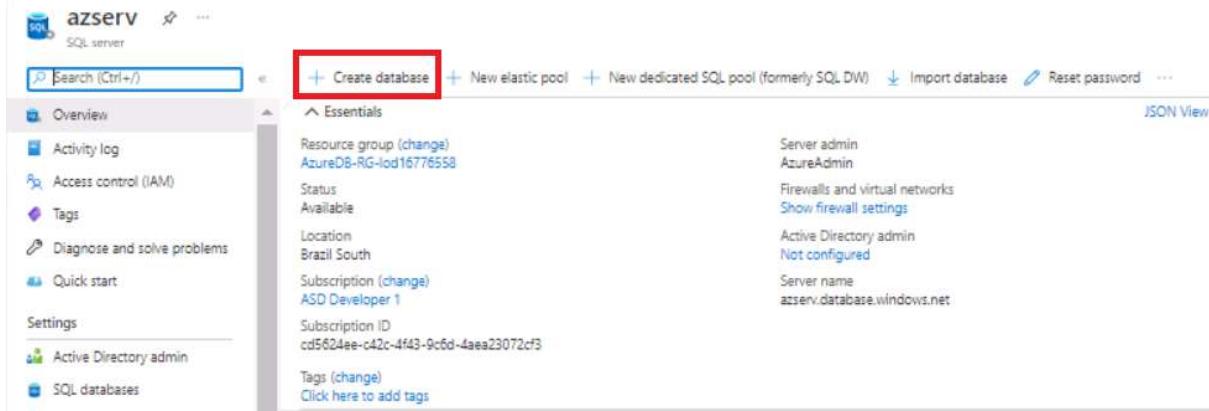
Open web browser and navigate to **T https://portal.azure.com** to connect to Microsoft Azure Portal. Login as **T @lab.CloudPortalCredential(LabUser).Username** using **T @lab.CloudPortalCredential(LabUser).Password** as the password.

- 3. Navigate to your logical SQL Server

Navigate to your logical SQL Server in the Azure Portal and select the Overview

- 4. Add a new database

Click **Create Database** to add a new SQL Database



- 5. Configure the database

Enter the following information to create the Azure SQL Database. Then click **Review + create**. After the review completes, click **Create**.

Setting	Value
Database name	<b>T AdventureWorksLT</b>
Server	<b>T &lt;LogicalServerName&gt;</b> (Check if this server is the primary)

Setting	Value
Want to use SQL elastic pool?	No
Compute + storage	Select <b>Configure database</b> , then <b>Basic</b> tile and click <b>Apply</b>
Backup storage redundancy	Geo-redundant backup storage
Data Source	In <b>Additional settings</b> , choose <b>Use existing data</b> and then <b>Sample</b>

**i** The **Data Source** settings are located under the **Additional Settings** tab.

Basics Networking Security Additional settings Tags Review + create

Customize additional configuration parameters including collation & sample data.

**Data source**

Start with a blank database, restore from a backup or select sample data to populate your new database.

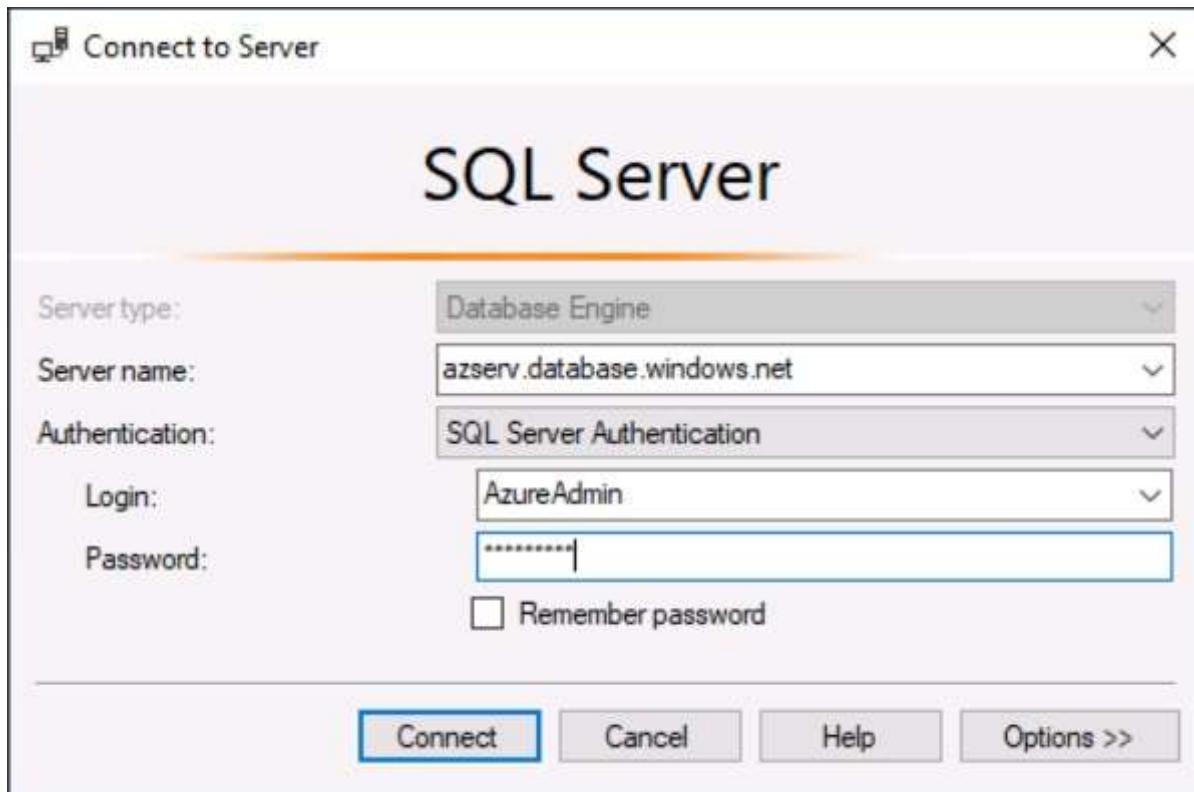
Use existing data \*  None  Backup  Sample

AdventureWorksLT will be created as the sample database.

## □ 6. Connect to Logical Server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
Server name	<input type="text"/> <LogicalServerName>.database.windows.net
Authentication	SQL Server Authentication
Login	<input type="text"/> AzureAdmin
Password	<input type="text"/> Passw0rd1



7. Expand the Columns

Open the **AdventureWorksLT** database, expand Tables then expand the **SalesLT.Customer** table and expand the columns.

The screenshot shows the SQL Server Object Explorer interface. At the top, it displays the connection information: azserv.database.windows.net (SQL Server 12.0.2000.8 - AzureAdmin). Below this, the tree structure of the database is shown:

- Databases
  - System Databases
  - AdventureWorksLT**
  - Database Diagrams
- Tables
  - System Tables
  - External Tables
  - GraphTables
  - dbo.BuildVersion**
  - dbo.ErrorLog**
  - SalesLT.Address**
  - SalesLT.Customer**
- Columns

The **SalesLT.Customer** table and its **Phone** column are highlighted with red boxes. The **Phone** column is also highlighted with a blue selection bar at the bottom of the list.

Column Name	Type	Properties
CustomerID	(PK, int, not null)	
NameStyle	(NameStyle(bit), not null)	
Title	(nvarchar(8), null)	
FirstName	(Name(nvarchar(50)), not null)	
MiddleName	(Name(nvarchar(50)), null)	
LastName	(Name(nvarchar(50)), not null)	
Suffix	(nvarchar(10), null)	
CompanyName	(nvarchar(128), null)	
SalesPerson	(nvarchar(256), null)	
EmailAddress	(nvarchar(50), null)	
<b>Phone</b>	<b>(Phone(nvarchar(25)), null)</b>	
PasswordHash	(varchar(128), not null)	
PasswordSalt	(varchar(10), not null)	
rowguid	(uniqueidentifier, not null)	

8. Select encrypt column

Right click on the **Phone** Column and Select **Encrypt Column**

The screenshot shows the Object Explorer in SQL Server Management Studio. A context menu is open over the 'Phone' column in the 'SalesLT.Customer' table. The menu items are: Encrypt Column..., Reports, Rename, Delete, and Refresh. The 'Encrypt Column...' item is highlighted with a red box.

- azserv.database.windows.net (SQL Server 12.0.2000.8 - AzureAdmin)
  - Databases
    - + System Databases
  - AdventureWorksLT
    - + Database Diagrams
  - Tables
    - + System Tables
    - + External Tables
    - + GraphTables
    - + dbo.BuildVersion
    - + dbo.ErrorLog
    - + SalesLT.Address
    - + SalesLT.Customer
  - Columns
    - + CustomerID (PK, int, not null)
    - + NameStyle (NameStyle(bit), not null)
    - + Title (nvarchar(8), null)
    - + FirstName (Name(nvarchar(50)), not null)
    - + MiddleName (Name(nvarchar(50)), null)
    - + LastName (Name(nvarchar(50)), not null)
    - + Suffix (nvarchar(10), null)
    - + CompanyName (nvarchar(128), null)
    - + SalesPerson (nvarchar(256), null)
    - + EmailAddress (nvarchar(50), null)
    - + **Phone (Phone(nvarchar(25)), null)**
    - + PasswordHash (varchar(128), not null)
    - + PasswordSalt (varchar(10), not null)
    - + rowguid (uniqueidentifier, not null)
    - + ModifiedDate (datetime, not null)

- 9. Move from Introduction Page

Click **Next** on **Introduction page** of **Always Encrypted** wizard

- 10. Configure Column Selection

Check the **Phone** column. Enter the following information. Click **Next**

Setting	Value
Name	Phone
State	No Change

Setting	Value
Encryption Type	Deterministic
Encryption Key	No Change

Always Encrypted

**Column Selection**

Introduction      Help

Column Selection      Master Key Configuration      Run Settings      Summary      Results

Search column name...

Apply one key to all checked columns: CEK\_Auto1 (New)

Name	State	Encryption Type	Encryption Key
SalesLT.Custo...		Deterministic	CEK_Auto1 (New)
Phone			

Show affected columns only

< Previous      Next >      Cancel

Always Encrypted

Column Selection

Search column name...

Apply one key to all checked columns: CEK\_Auto1 (New)

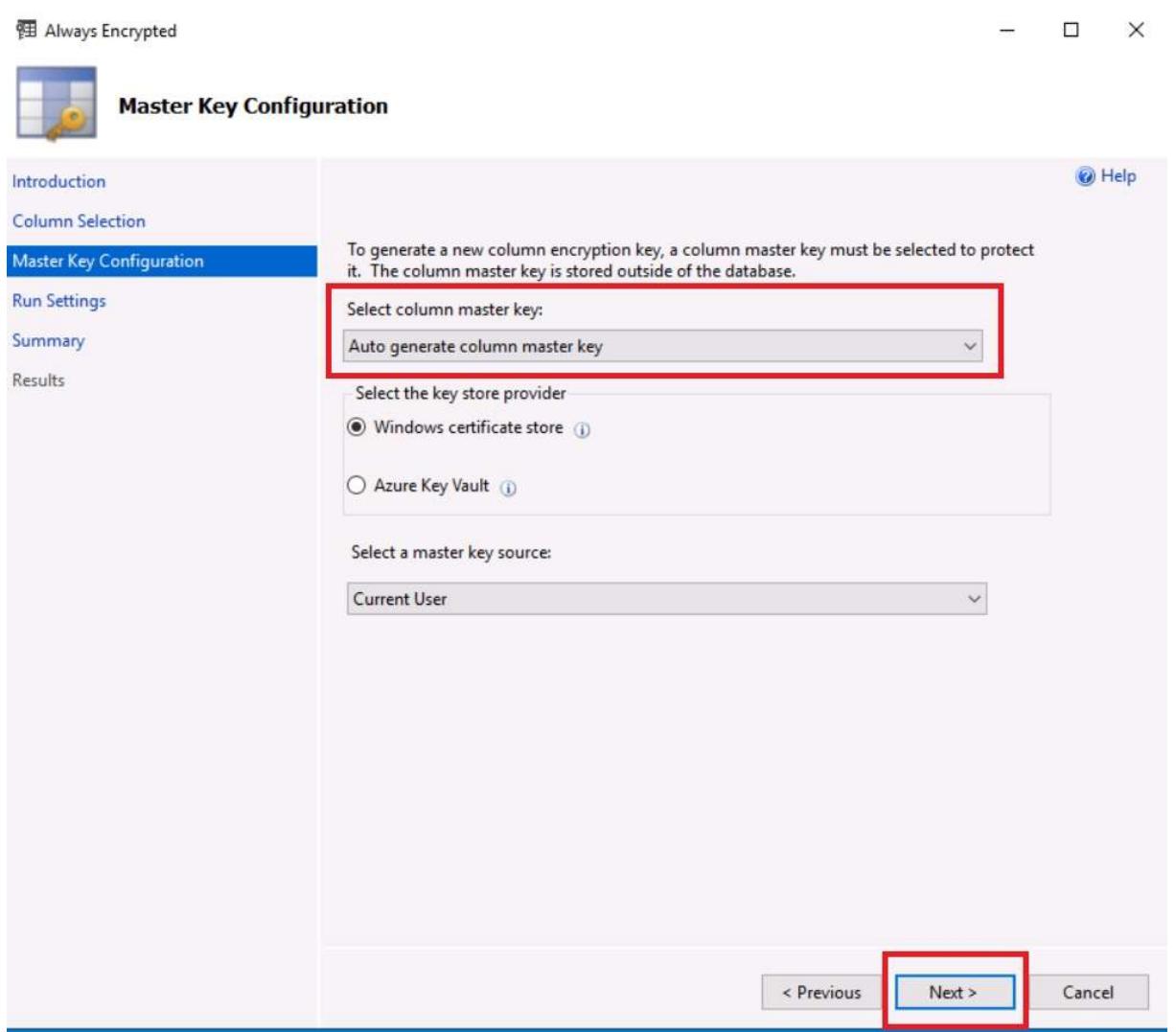
Name	State	Encryption Type	Encryption Key
SalesLT.Custo...		Deterministic	CEK_Auto1 (New)
Phone			

Show affected columns only

< Previous      Next >      Cancel

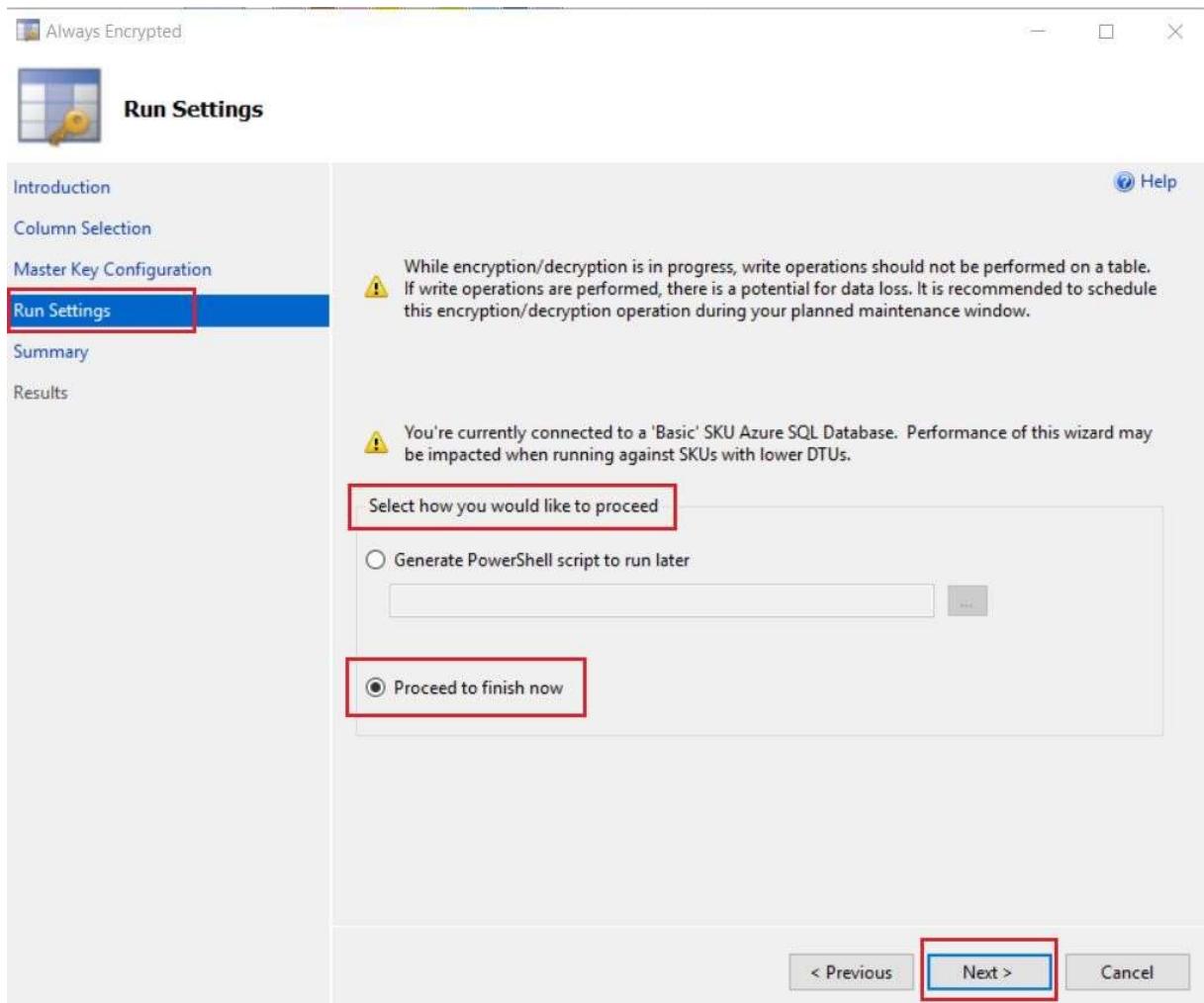
11. Configure Master Key Configuration

Change master key to Auto Generate column master key if not and click **Next**



#### 12. Configure Run Settings Configuration

Under "Select how would you like to proceed", validate you checked the option "Proceed to finish now" and click **Next**.



13. Complete Always Encrypted

Click **Finish** to complete Always Encrypted configuration and wait until the encryption finishes then click **Close** to close the wizard.

**i** This may take a few minutes.



## Summary

[Introduction](#)[Column Selection](#)[Master Key Configuration](#)[Run Settings](#)[Summary](#)[Results](#)[Help](#)**Verify the choices made in this wizard.**

Click Finish to perform the operations with the following settings:

- Source database settings
  - Source server name: azserv.database.windows.net
  - Source database name: AdventureWorksLT
- Create new master key
  - New master key name: CMK\_Auto1
  - New master key in windows certificate store:\CurrentUser
- Create new encryption key
  - New encryption key: CEK\_Auto1
- Encrypt column Phone
  - Table name: Customer
  - Encryption key name: CEK\_Auto1
  - Encryption type: Deterministic

[< Previous](#)[Finish](#)[Cancel](#)

## Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Use the .Net Client App to Test Always Encrypted

This exercise shows how to test Always Encrypted

### Tasks

- 1. Query encrypted column

Within **SSMS**, right-click the **AdventureWorksLT** database, choose **New Query** and **Execute** the following query.

SQL

▶ `SELECT Title, FirstName, LastName, CompanyName, Phone FROM [SalesLT].[Customer]`

- 2. Check the Phone column in the result

The Phone column in the query result is encrypted

	Title	FirstName	LastName	CompanyName	Phone
1	Mr.	Orlando	Gee	A Bike Store	0x014A122A2B9D889D91C4AA6E08B54EE404DB8BEC175A78...
2	Mr.	Keith	Harris	Progressive Sports	0x01FEF01C38F5E3A25831CA6E0B04E2200100B1DB6D69F4BB...
3	Ms.	Donna	Carreras	Advanced Bike Components	0x01FD6CD45B898525251A16E3A351F008DB978B463A89FD5...
4	Ms.	Janet	Gates	Modular Cycle Systems	0x016786A6153CD76A243EB8D773F4BC7EA7C06DBC8C89CA41...
5	Mr.	Lucy	Harrington	Metropolitan Sports Supply	0x0161DDCC3810387605CAF7F1346FC07168453ABDEEE4A8...
6	Ms.	Rosmarie	Carroll	Aerobic Exercise Company	0x014A888E54D22549160730B6530EC5F25758AC027C2A23E03...
7	Mr.	Dominic	Gash	Associated Bikes	0x01E1D5B5CE51E091A70FADBB6CD1C4F4531872EAF24A744...
8	Ms.	Kathleen	Garza	Rural Cycle Emporium	0x01BD0502D8DCEA6C5ACA84DA6DB3C31F6DE85BF51CA8FD...
9	Ms.	Katherine	Harding	Sharp Bikes	0x018DE5BF5AA38421168592F674EAB2C60D39B870987196...
10	Mr.	Johnny	Caprio	Bikes and Motorbikes	0x0105AE2BEB8EC5FAF1896B2728E89A00C317FA59CF793CCD...
11	Mr.	Christopher	Beck	Bulk Discount Store	0x01C49398D8B7E505BD8CE52E7A94A82583E8FC7DD5133D7...
12	Mr.	David	Liu	Catalog Store	0x01CBADCBAA49BB040A59FC9C2AA4CF1084D0F2A4E49292...
13	Mr.	John	Beaver	Center Cycle Shop	0x012F5841E83E0CDAD974B0D33A6F1F8C1E28B7887A61B2C...
14	Ms.	Jean	Handley	Central Discount Store	0x0144E7072E9F5C83959085F27C2AEADC6657F6433452FE5D...
15	N...	Jinghao	Liu	Chic Department Stores	0x01F0A4B4E86695EC693A159FE1C1108C71F8FB23ADC800B968...
16	Ms.	Linda	Burnett	Travel Systems	0x01B2C438053E8EF27819050A938694B3C6E34B3A04DA965...
17	Mr.	Kerim	Hanif	Bike World	0x0136CF522ABA14696FCA328D8D73E9487ABD8D80896123C...
18	Mr.	Kevin	Liu	Eastside Department Store	0x013E9AA87729677C84FF992C6424F745FA2183D92098ED00...
19	Mr.	Donald	Blanton	Coalition Bike Company	0x01A6773C9F5088A18097C4BCC96C876730CB871E52041BC7A...
20	Ms.	Jackie	Blackwell	Commuter Bicycle Store	0x0134BD6A3047A5C1541F88DB61B86D3C51B627840843807...
21	Mr.	Bryan	Hamilton	Cross-Country Riding Supp...	0x014CC2D3C0E68958AD347157A8DBB27E4F2BCCAB19A6E13...

- 3. Run AlwaysEncryptedClient.exe

In the **C:\LabFiles\M04L06Lab01\AlwaysEncryptedClient** folder, run **AlwaysEncryptedClient.exe**, ensure that the connection string and SQL command is correct. Click the **Get Data** button and verify Phone column is not encrypted.

- i** In the connection string box change the **Data Source** from **azserv.database.windows.net** to the unique name of the server you created in the first lab.
  
- File Explorer may **hide known extensions** so make sure you run the **AlwaysEncryptedClient** listed as an **Application**. (Not the Visual Studio solution file)

Always Encrypted Demo Client

Connection String

```
Data Source=azserv.database.windows.net;Initial Catalog=AdventureWorksLT;User ID=AzureAdmin;Password=Passw0rd1;Column Encryption Setting=Enabled
```

Select Title, FirstName, LastName, CompanyName, Phone from [SalesLT].[Customer]

	Title	FirstName	LastName	CompanyName	Phone
▶	Mr.	Orlando	Gee	A Bike Store	245-555-0173
	Mr.	Keith	Harris	Progressive Sports	170-555-0127
	Ms.	Donna	Cameras	Advanced Bike Components	279-555-0130
	Ms.	Janet	Gates	Modular Cycle Systems	710-555-0173
	Mr.	Lucy	Harrington	Metropolitan Sports Supply	828-555-0186
	Ms.	Rosmarie	Carroll	Aerobic Exercise Company	244-555-0112
	Mr.	Dominic	Gash	Associated Bikes	192-555-0173
	Ms.	Kathleen	Garza	Rural Cycle Emporium	150-555-0127
	Ms.	Katherine	Harding	Sharp Bikes	926-555-0159
	Mr.	Johnny	Caprio	Bikes and Motorbikes	112-555-0191
	Mr.	Christopher	Beck	Bulk Discount Store	1 (11) 500 555-0132

- 4. Close all open windows without saving.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab

# Implement Row Level Security

---

## Introduction

In this lab, you will learn how to implement Row Level Security.

## Estimated Time

10 minutes

## Objectives

After completing this lab, you will be able to:

- Implement Row Level Security
- Define a security policy that filters the rows of a table
- Implement block predicates

## Logon Information

1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `[T]`  
`@lab.VirtualMachine(AzureSQLDB).Username` using `[T]`  
`@lab.VirtualMachine(AzureSQLDB).Password` as the password.

Click the `[T] Type Text` icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Implement Row Level Security on Azure SQL Database

---

This exercise shows how to implement Row Level Security.

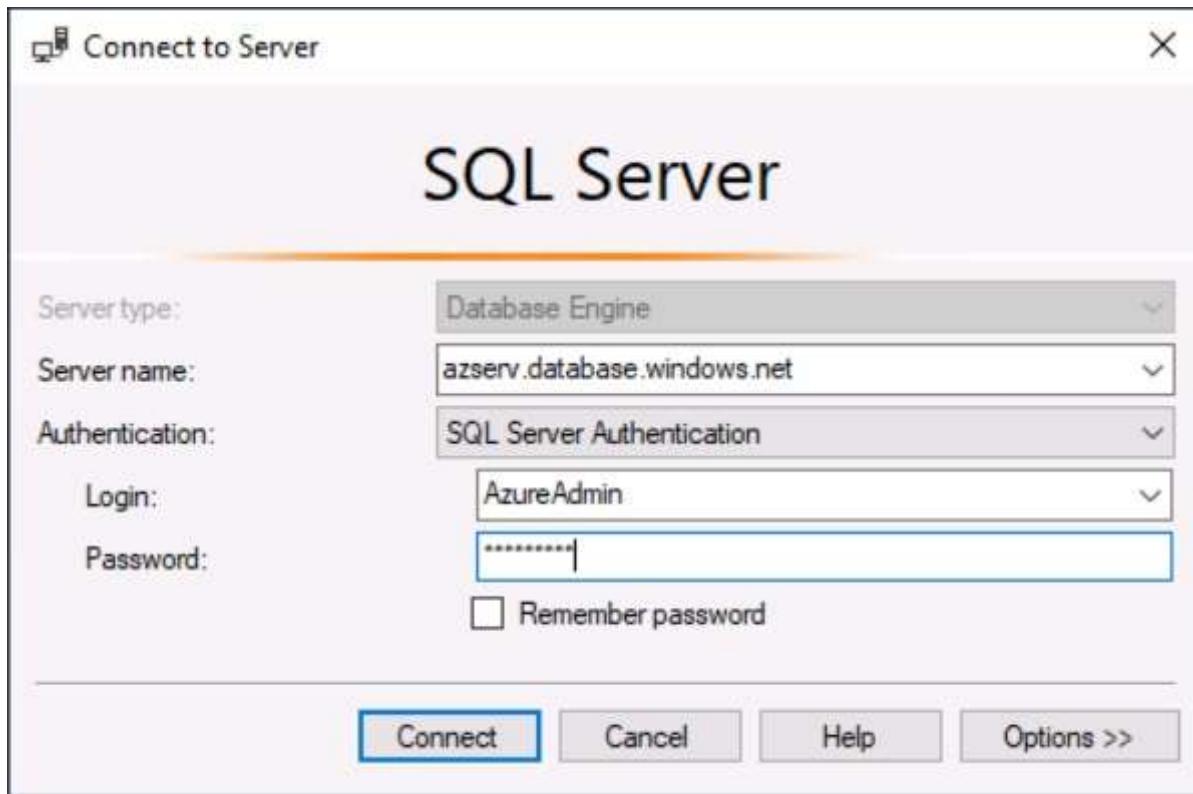
---

## Tasks

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB.Username` using  `@lab.VirtualMachine(AzureSQLDB.Password` as the password.
- 2. Connect to Logical Server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the following information to connect to Azure logical server. Click **Connect**.

Setting	Value
<b>Server name</b>	<input type="text"/> <LogicalServerName>.database.windows.net
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text"/> AzureAdmin
<b>Password</b>	<input type="text"/> Passw0rd1



□ 3. Open a new Query Window on the **Salesdb**

□ Right click on the **Salesdb** and select **New Query**.

**i** If you completed a previous lab that created a Failover Group and performed a failover, make sure you select the current primary logical SQL server otherwise you will receive an error indicating that the **Salesdb is read-only** indicating you selected the secondary replica.

□ 4. Execute the script

Open the script **C:\LabFiles\M04L07Lab01\RowLevelSecuritySetUp.sql** by using the **File/Open/File** menu option or the **Open File** icon on the SSMS Toolbar. After opening the script, execute the step by step instructions.

**i** If you receive a Message 111 that CREATE SCHEMA must be first in a query batch. Look inside script and type **GO** after step 3 **DENY UPDATE** and before step 4, **CREATE SCHEMA**

□ 5. Open a new Query Window on the **Salesdb** and execute the following script to simulate the connection filtering by selecting from the Sales table after setting different user IDs in SESSION\_CONTEXT:

**i** In practice, the application is responsible for setting the current user ID in SESSION\_CONTEXT after opening a connection.

SQL

```
▶ EXECUTE AS USER = 'AppUser';
EXEC sp_set_session_context @key=N'UserId', @value=1;
SELECT COUNT(*) FROM SalesLab;
SELECT * FROM SalesLab;
GO
```

- 6. In the same window execute the following query, to insert a new row to the table:

SQL

```
▶ INSERT INTO SalesLab VALUES (7, 1, 'Seat', 12); -- inserting row as it is
GO
```

- 7. In the same window execute the following query, to insert another row to the table:

SQL

```
▶ INSERT INTO SalesLab VALUES (8, 2, 'Wheel', 2); -- error: blocked from inser
GO
```

- 8. In the same window execute the following query, simulate another connection with the correct user ID and try to insert a row:

SQL

```
▶ EXECUTE sp_set_session_context @key=N'UserId', @value=2;
INSERT INTO SalesLab VALUES (8, 2, 'Wheel', 2); -- inserting row as it is
GO
```

**i** The all above queries are available in  
**C:\LabFiles\M04L07Lab01\RowLevelSecurityBlockPredicate.sql** file.

- 9. Now we are simulating the update option. Open a new Query Window on the **Salesdb** and execute the following query:

SQL

```
▶ EXEC sp_set_session_context @key=N'UserId', @value=2;
UPDATE SalesLab SET AppUserId=1 where OrderId=8 -- error: update permission
```

- 10. In the same window execute the following query, if we try to update another column:

SQL

```
▶ UPDATE SalesLab SET Product='Valve' where OrderId=8 -- it works as the Appl  
GO
```

**i** The all above queries are available in  
**C:\LabFiles\M04L07Lab01\RowLevelSecurityUpdate.sql** file.

- 11. Close all open windows.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Vulnerability Assessment

---

## Introduction

In this lab, you will learn how to connect to run a Vulnerability Assessment, Review the report and set a baseline

## Estimated Time

20 minutes

## Objectives

After completing this lab, you will be able to:

- Create a Vulnerability Assessment
- Review the report of the Vulnerability Assessment
- Set a baseline based on the Vulnerability Assessment

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as  **@lab.VirtualMachine(AzureSQLDB).Username** using  **@lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Create a new database AdventureWorksLT

---

This exercise shows how to create a new database AdventureWorksLT

- To make a Vulnerability Assessment, we will use another test database **AdventureWorksLT**. If this database already exists on your logical server, go to Exercise 2.

## Tasks

- 1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as  **@lab.VirtualMachine(AzureSQLDB).Username** using  **@lab.VirtualMachine(AzureSQLDB).Password** as the password.
- 2. Connect to the Azure Portal and browse to your SQL Server (logical server) **<LogicalServerName>**.

Open the Azure Portal in your browser  <https://portal.azure.com>

Login as  **@lab.CloudPortalCredential(LabUser).Username** using  **@lab.CloudPortalCredential(LabUser).Password** as the password.

- 3. Configure a new database

Navigate to your logical SQL server **<LogicalServerName>**, then click on the button **New Database**

**i** If you have created **AdventureWorksLT** in previous exercise, skip to Exercise 2

Specify the following options of the new database

Option	Value
<b>Database name</b>	<input type="text"/> <b>AdventureWorksLT</b>
<b>Server</b>	<input type="text"/> <b>&lt;LogicalServerName&gt;</b> (Check if this server is the primary)
<b>Want to use SQL elastic pool?</b>	<b>No</b>
<b>Compute + storage</b>	Select <b>Configure database</b> , then <b>Basic</b> tile and click <b>Apply</b>
<b>Backup storage redundancy</b>	Geo-redundant backup storage

Option	Value
Data Source	In <b>Additional settings</b> , choose <b>Use existing data</b> and then <b>Sample</b>

[!NOTE] The **Data Souce** and **Collation** settings are located under the **Additional Settings** tab.

- 1. Click the **OK** button

Wait until the database is created

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Run a scan

This exercise shows how to run a scan

### Tasks

- 1. Run a Vulnerability Assessment

Open the Azure Portal in your browser <https://portal.azure.com>

Login as [@lab.CloudPortalCredential\(LabUser\).Username](#) using [@lab.CloudPortalCredential\(LabUser\).Password](#) as the password.

Navigate to the **SQL database** that you created earlier, named **AdventureWorksLT**

Click **Overview** and then **Microsoft Defender for SQL**.

Click to **Enable Microsoft Defender for SQL**, and wait for the feature to be enabled. After that **Vulnerability Assessment findings** will populate. Then click on the **View additional findings in Vulnerability Assessment**, which automatically opens the Vulnerability Assessment settings card.

- You can click on any issue listed to get more detailed information on that issue.

Home > AdventureWorksLT (azserv/AdventureWorksLT)

AdventureWorksLT (azserv/AdventureWorksLT) | Security Center

No recommendations to display

There are no security recommendations for this resource

View all recommendations in Security Center

Security incidents and alerts

Security Center uses advanced analytics and global threat intelligence to alert you to malicious activity. Alerts displayed below are from the past 21 days.

Check for Azure Defender Alerts on this resource in Azure Security Center >

Vulnerability assessment findings

ID	Security Check	Applies to	Severity
No results			

[View additional findings in Vulnerability Assessment >](#)

- 2. Run a scan

Click on the **Scan** button

Wait until the Vulnerability Assessment scan is finished.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 3: View the report

### Tasks

- 1. View the Vulnerability Assessment Dashboard

When your scan is complete, your scan report is automatically displayed in the Azure portal.

- The report presents an overview of your security state: how many issues were found and their respective severities. You should see multiple issues that apply to AdventureWorksLT

Home > AdventureWorksLT (azserv/AdventureWorksLT) >

## Vulnerability Assessment

(azserv/AdventureWorksLT)

Scan Export Scan Results Scan History Feedback

SQL Vulnerability Assessment rules have been updated. This may impact your scan results. Learn more →

Total failing checks: 6 (Red) Total passing checks: 28 (Green)

Risk	Count
High Risk	3
Medium	2
Low Risk	1

Last scan time: Wed, 12 May 2021 00:31:50 UTC

Learn more

SQL Vulnerability Assessment

**Findings (6)**

ID	SECURITY CHECK	APPLIES TO	CATEGORY	RISK	ADDITIONAL INFO
VA2108	Minimal set of principals should be members of fixed high i...	AdventureWorksLT	Authentication & Au...	High	Should be
VA2061	Auditing should be enabled at the server level	master	Auditing & Logging	High	Should be
VA2065	Server-level firewall rules should be tracked and maintained ...	master	Surface area reduction	High	Should be
VA1143	'dbo' user should not be used for normal service operation	AdventureWorksLT	Surface area reduction	Medium	

**Passed (28)**

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 4: Analyze and resolve

---

This exercise shows how to analyze and resolve

### Tasks

Set as Baseline and rerun the scan

- 1. Click on the first issue
- 2. Explore the details of the issue
- 3. Click on **Approve as Baseline** and click **Yes** to confirm.
- 4. Repeat steps 1 - 3 for the second issue
- 5. Run the scan again.

**i** You will notice that the 2 issues are not in list of failing checks anymore

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 5: Review the scan history

---

This exercise shows how to review the scan history

### Tasks

- 1. Click the **Scan History** Button

**i** Notice that there are multiple scans in the scan history

- 2. Click on the first Scan and the **Vulnerability Assessment Report** will open.
- 3. Click on **Export Scan Results** and then **Download** to export the report to Excel.

Excel is not installed on the virtual machine so if you try to open the downloaded file it will fail.

- 4. Close all open windows without saving.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Monitoring Query Performance using Query Performance Insights

---

## Introduction

In this lab, you will learn how to connect to monitor Azure SQL Database query performance using Query Performance Insights

## Estimated Time

15 minutes

## Objectives

After completing this lab, you will be able to:

- Monitor Azure SQL Database using Query Performance Insights

## Logon Information

1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB).Username` using  `@lab.VirtualMachine(AzureSQLDB).Password` as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Monitor Query Performance using Query Performance insights

---

This exercise shows how to monitor Azure SQL Database query performance using Query Performance Insights.

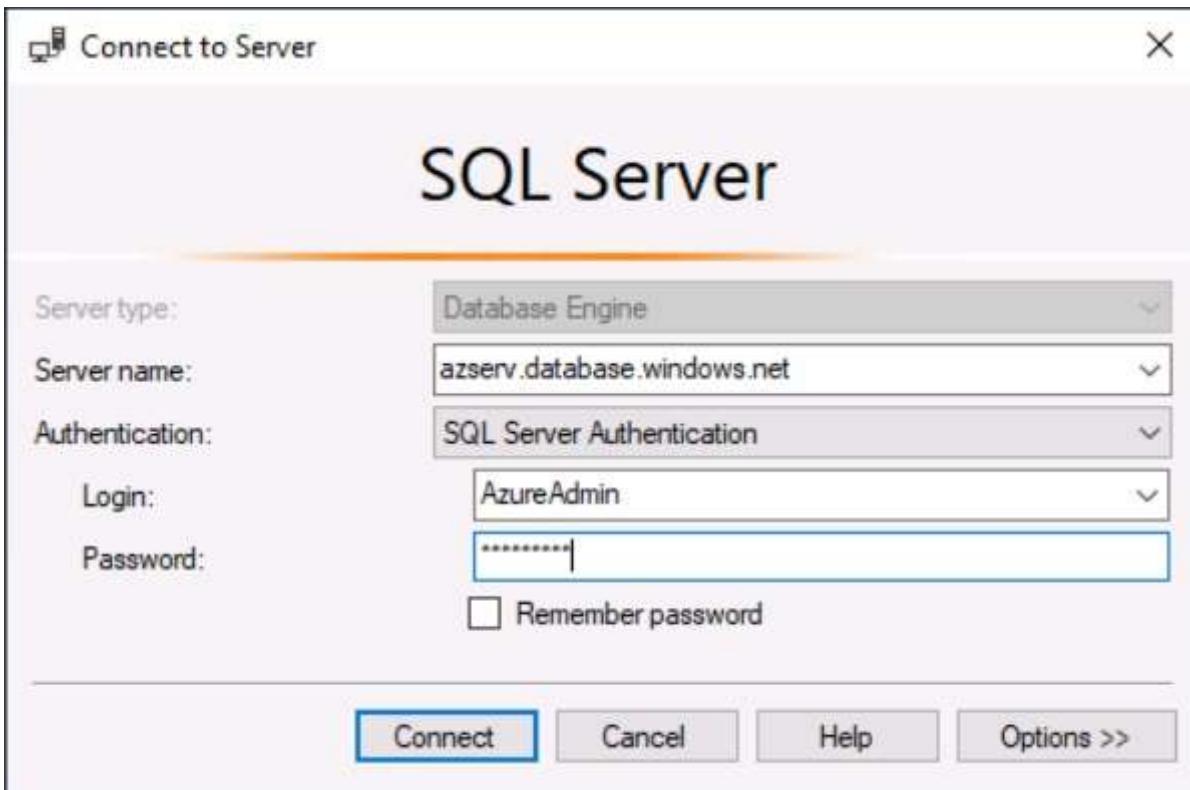
---

## Tasks

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `T @lab.VirtualMachine(AzureSQLDB).Username` using `T @lab.VirtualMachine(AzureSQLDB).Password` as the password.
- 2. Connect to Logical Server with SSMS

Open SQL Server Management Studio (**SSMS**). Enter the following information to connect to Azure logical server. Click **Connect**.

Settings	Values
<b>Server name</b>	<code>T &lt;LogicalServerName&gt;.database.windows.net</code>
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<code>T AzureAdmin</code>
<b>Password</b>	<code>T Passw0rd1</code>



### 3. Enable Query Store

Right-click on the **AdventureWorksLT** database and Select **Properties** and go to the **Query Store** page. Set **Operation Mode (Requested)** to **Read Write**. This should already have been set by default.

- The setting for Query Store must be enabled to leverage Query Performance Insights.

#### **Operation Mode**

- **Off:** Will turn off the Query Store.
- **Read Only and Read Write:** The different modes in which you can have the Query Store, when it is on. In the READ\_WRITE mode, the query store collects and persists the query plan and runtime execution statistics information. In the READ\_ONLY mode, information can be read from the query store, but new information is not added. If the maximum allocated space of the query store is exhausted, the query store will change its operation mode to READ\_ONLY.

### 4. Set Data Flush Interval

Change data flush interval to **1** for the lab purposes.

- Data Flush Interval (Minutes):** To optimize for performance, data collected by the Query Store is asynchronously written to the disk. This property configures the frequency at which query store data is flushed and persisted to disk.

### 5. Set Statistics Collection Interval

Change Statistics Collection Interval to **1 Minute** for lab purposes

- Statistics Collection Interval:** Gets and sets the statistics collection interval value.

- 6. Set Max Size (MB)

Keep **Max Size (MB)** as default (100)

- Max Size (MB):** Max size allocated to the query store for the database.

- 7. Set Query Store Capture Mode

Change **Query Store Capture Mode** to **All** for lab purposes

- Query Store Capture Mode:**

- All:** Capture all queries
- Auto:** Capture based on resource consumption
- None:** Does not capture any new queries

- 8. Set Size Based Cleanup Mode

Keep **Size Based Cleanup Mode** as default (**Auto**)

- Size Based Cleanup Mode:**

- Off:** Data cleanup never happens
- Auto:** Data cleanup happens automatically

- 9. Set Stale Query Threshold (Days)

Keep **Stale Query Threshold (Days)** as default (**7**)

- Stale Query Threshold (Days):** Used to specify the number of days to retain data in the query store

- 10. Review Pie Charts

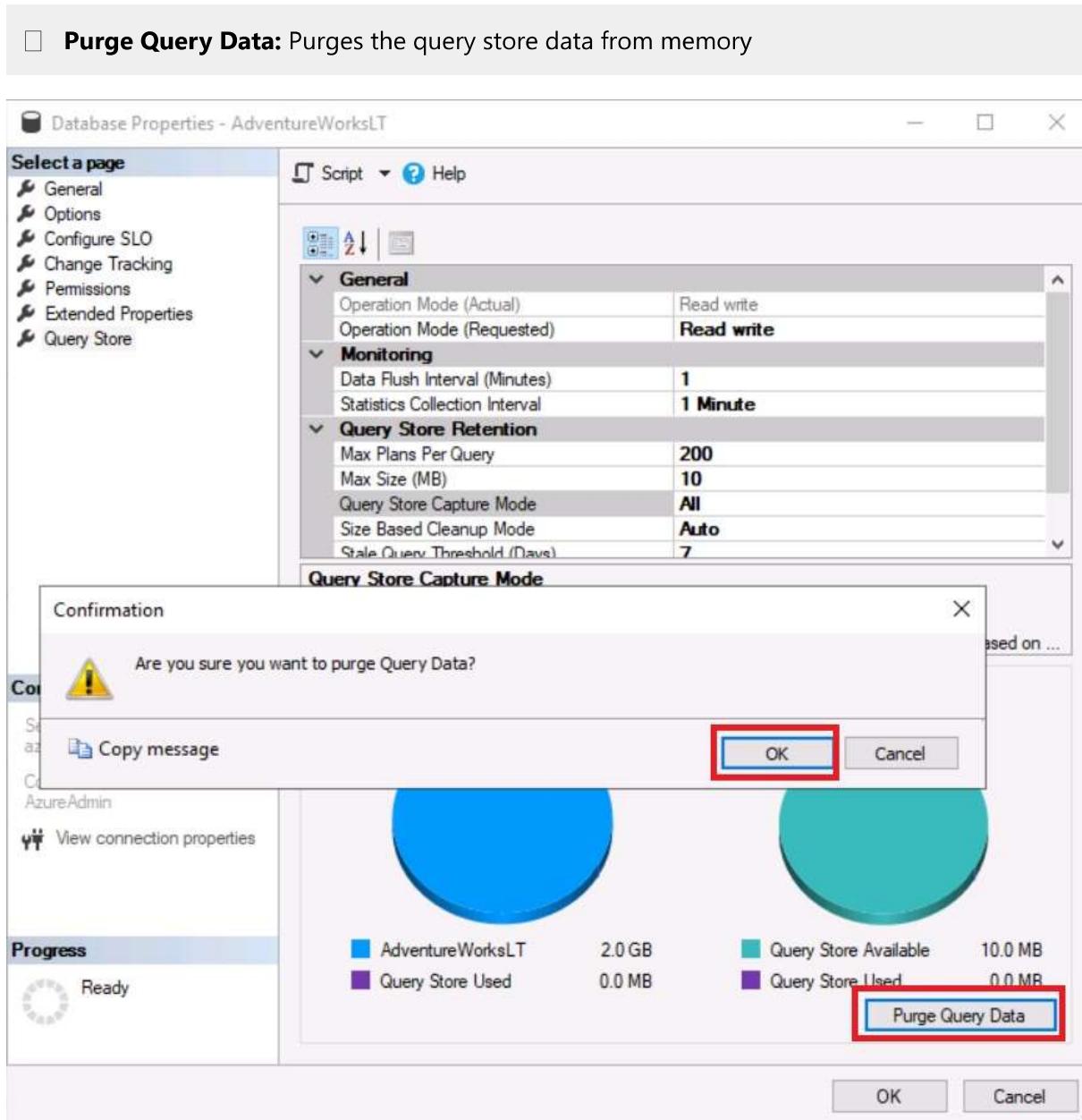
**Review** Pie Charts info.

- Pie Charts**

- The left chart** shows the total database file consumption on the disk, and the portion of the file which is filled with the query store data.
- The right chart** shows the portion of the query store quota which is currently used up. Note that the quota is not shown in the left chart. The quota may exceed the current size of the database

- 11. Click Purge Query Data with GUI

- Click **Purge Query Data** button, click **OK** to confirm, then click **OK** to save your changes.



12. Run a workload

- a. Open a new Query Window

Open a **new query** window and execute the query to generate some test load on the **AdventureWorksLT** database.

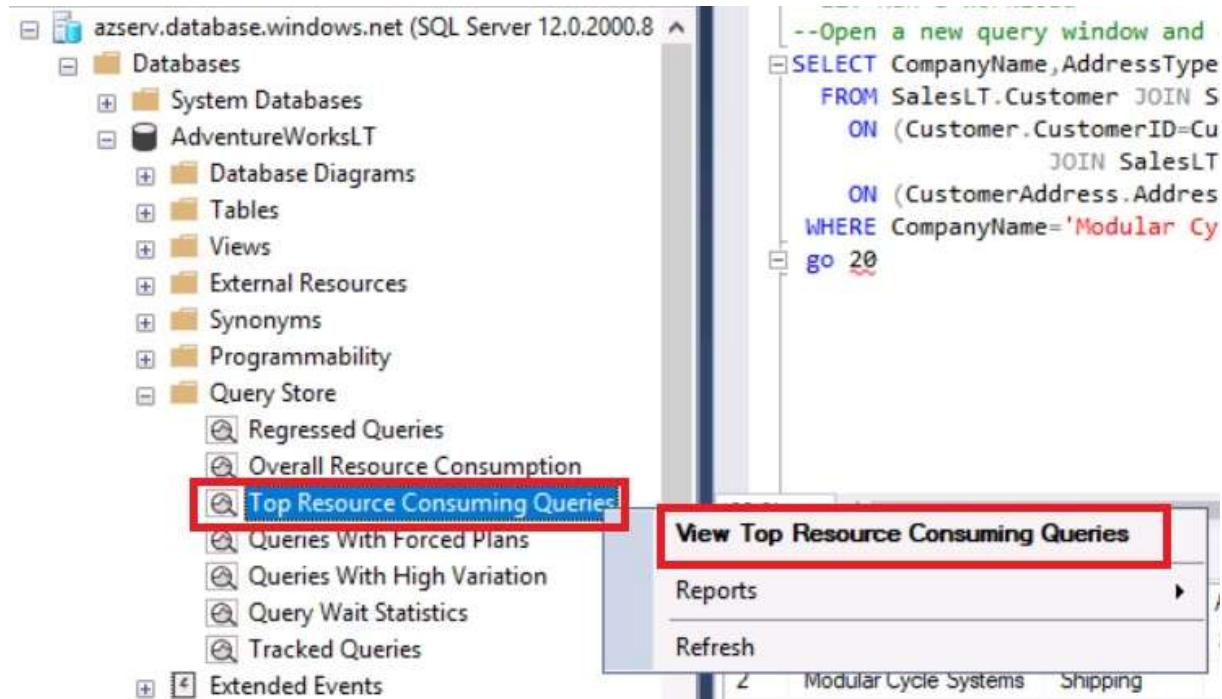
- b. Execute the script

Open the script **C:\LabFiles\M05L02Lab01\16918\_M05\_L02\_Lab.sql** by using the **File/Open/File** menu option or the **Open File** icon on the SSMS Toolbar. After opening the script, click **Execute** only step 12.

**i** You can ignore any intellisense errors.

13. Check top resource consuming queries

Check top resource consuming queries. Expand **AdventureWorksLT**, Expand **Query Store**, right-click on the "Top Resource Consuming Queries" and select "View Top Resource Consuming Queries".



Once the dashboard is open you will see your query as one of the Top consuming queries. Verify the query plan.

- Your query may not be the highest bar in the Histogram. Some system queries may be higher duration. Click on each bar in the Histogram until you identify your query plan.

14. Add an extra index

The query plan of the query you executed suggests creating a new index on the column **CompanyName**. Execute the script below to create the index:

```
SQL
▶ CREATE NONCLUSTERED INDEX [IX_CompanyName] ON [SalesLT].[Customer] ([Compa
```

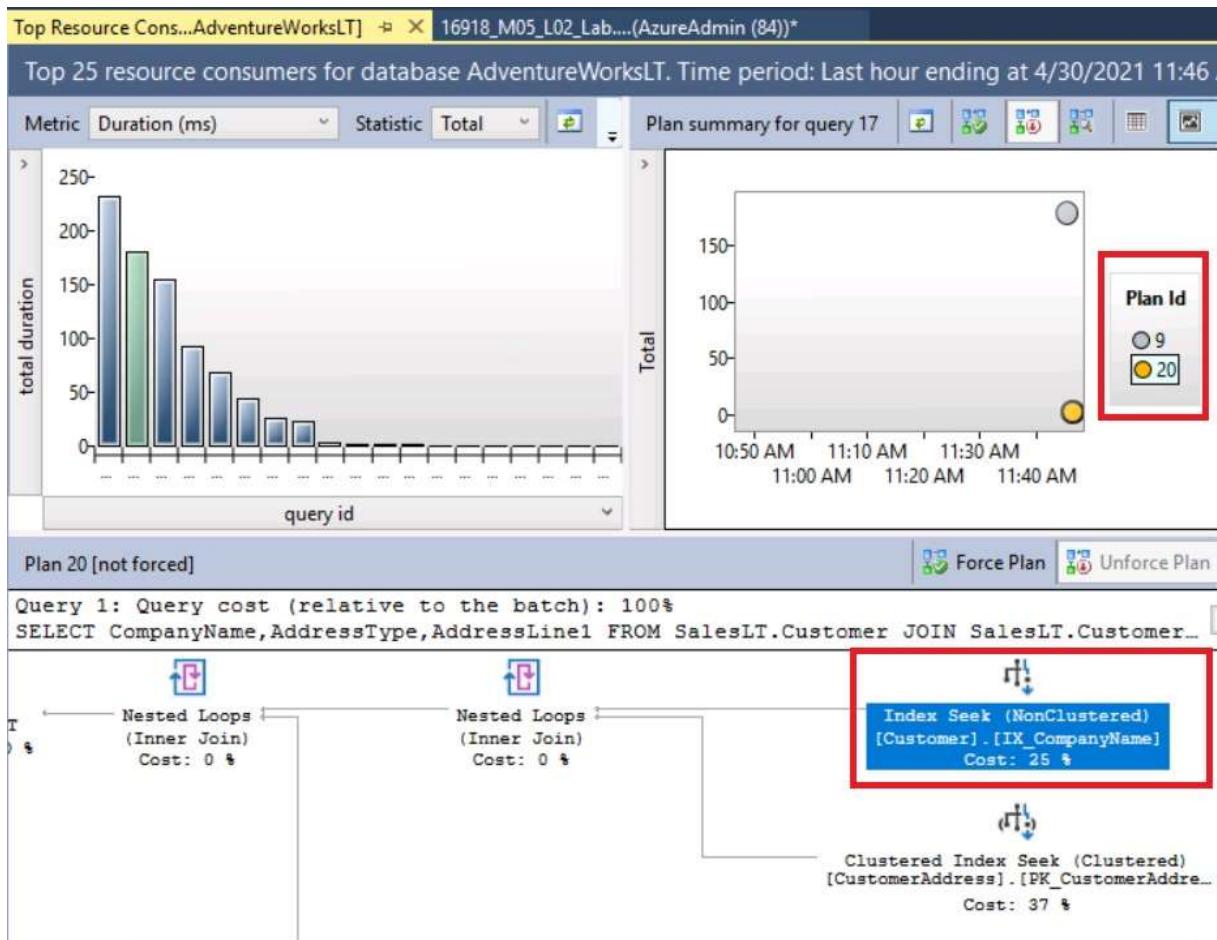
15. Run the same workload again

Run the same workload query again as before.

16. Check top resource consuming queries

Go back to the Top Resource Consuming Queries of the Query Store and click the refresh button.

- You will notice that your query has now 2 query plans. One that is using a Clustered Index Scan and one that is using the Index seek of the index that we have created after the first execution.



17. Connect to Microsoft Azure Portal

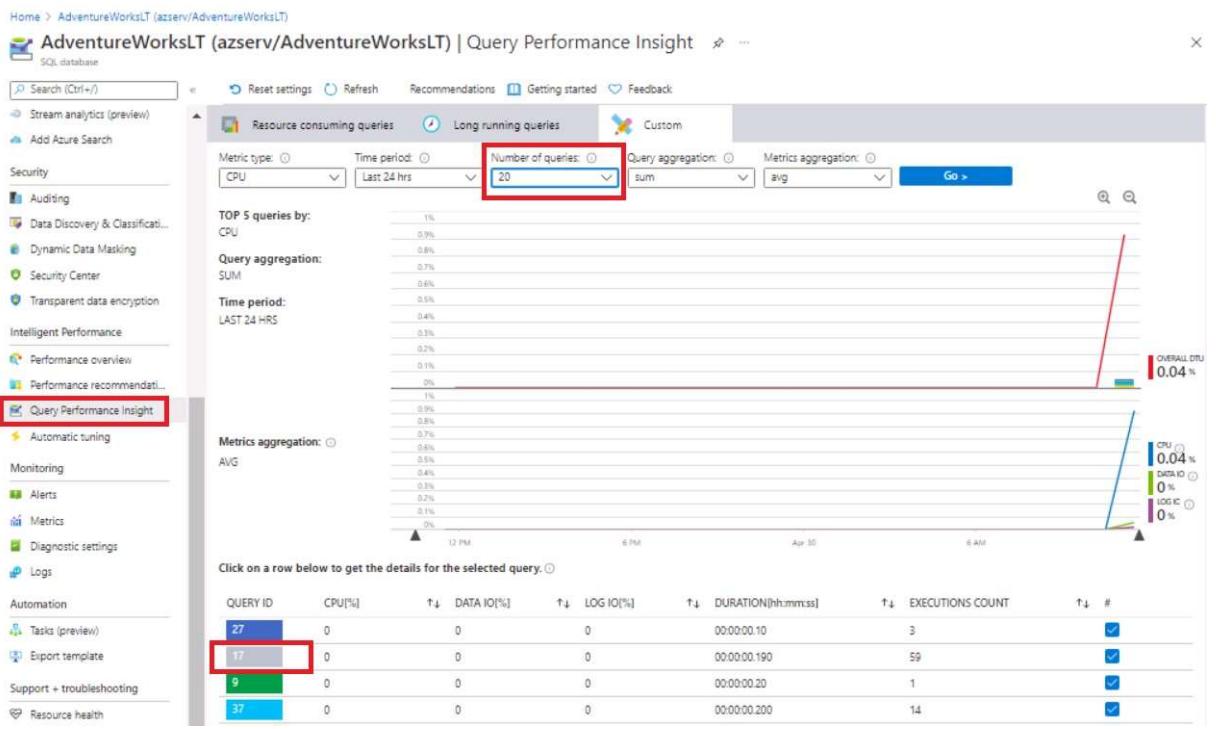
Open Internet Explorer and navigate to <https://portal.azure.com> to connect to **Microsoft Azure Portal**. Login as **@lab.CloudPortalCredential(LabUser).Username** using **@lab.CloudPortalCredential(LabUser).Password** as the password.

18. Check the Query Performance

Check the query performance through azure portal. Go to **AdventureWorksLT** database and Click **Query Performance Insight**

Search the query ID that you have identified with the query store and select the query details.

- Under the **Custom** tab, you may have to change the **Number of queries** from 5 to **20** if the query ID you identified is not listed in the Top 5 list.



Close all open windows without saving.

Congratulations!

You have successfully completed this lab. Click **Next** to advance to the next lab.

# Monitoring Azure SQL Database Performance using Extended Events

---

## Introduction

In this lab, you will learn how to monitor Azure SQL Database Performance using Extended Events.

## Estimated Time

10 minutes

## Objectives

After completing this lab, you will be able to:

- Monitor Azure SQL Database using extended events

## Logon Information

1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB).Username` using  `@lab.VirtualMachine(AzureSQLDB).Password` as the password.

Click the  **Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Monitor Azure SQL Database using Extended Events

---

This exercise shows how to monitor Azure SQL Database Performance using Extended Events.

---

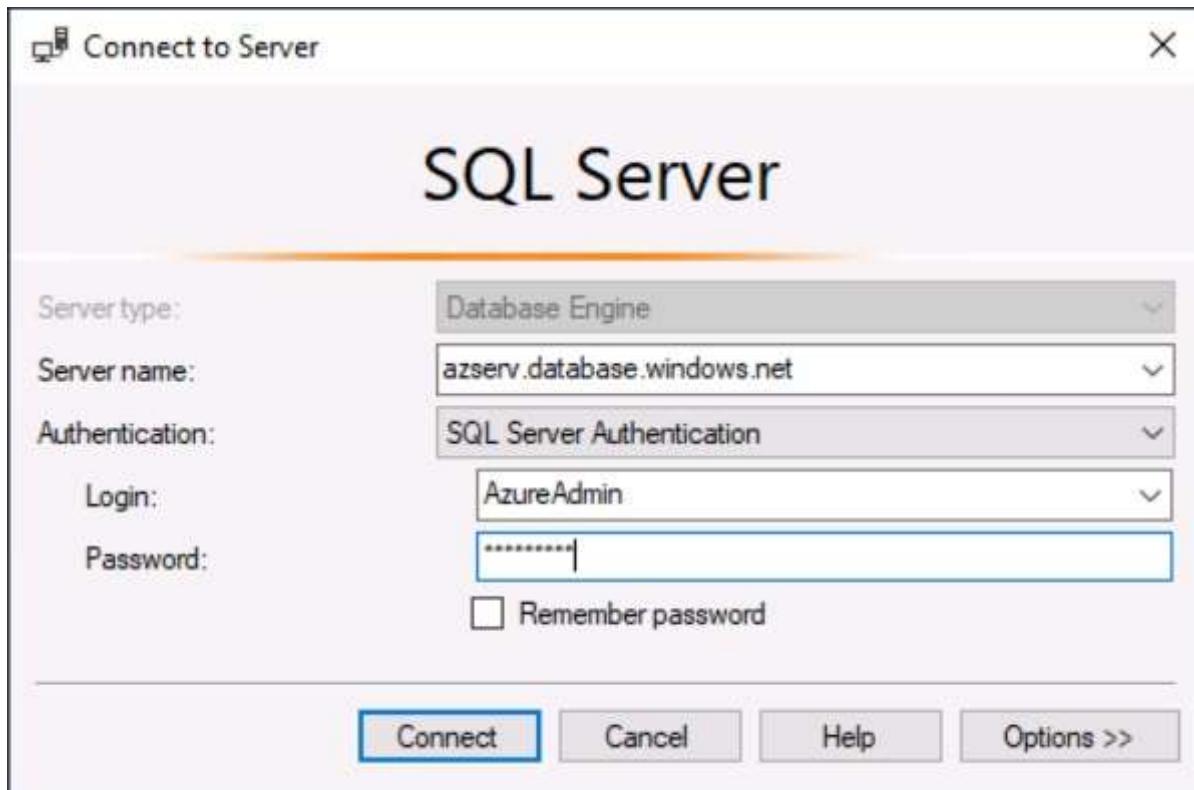
## Tasks

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB.Username` using  `@lab.VirtualMachine(AzureSQLDB.Password` as the password.
- 2. Connect to **Logical Server** with **SSMS**

Open SQL Server Management Studio (**SSMS**). Enter the following information to connect to Azure logical server. Click **Connect**.

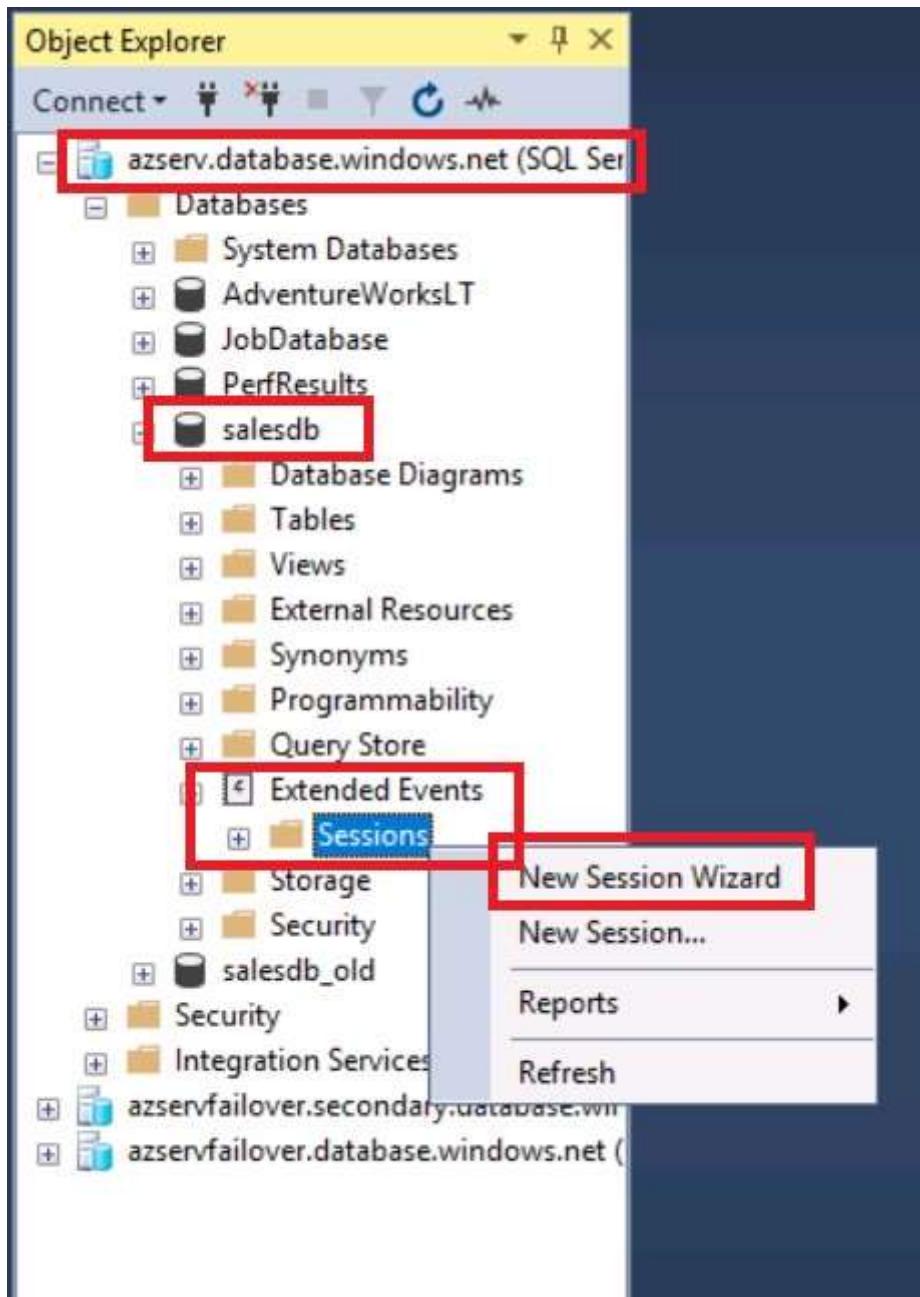
- If you completed a previous lab that creates a Failover Group and performs a failover, make sure you select the current primary logical SQL server otherwise you will receive an error later indicating that the Salesdb is read-only indicating the secondary replica.

Setting	Value
<b>Server name</b>	<code>&lt;LogicalServerName&gt;.database.windows.net</code>
<b>Authentication</b>	SQL Server Authentication
<b>Login</b>	<input type="text"/> <code>AzureAdmin</code>
<b>Password</b>	<input type="text"/> <code>Passw0rd1</code>



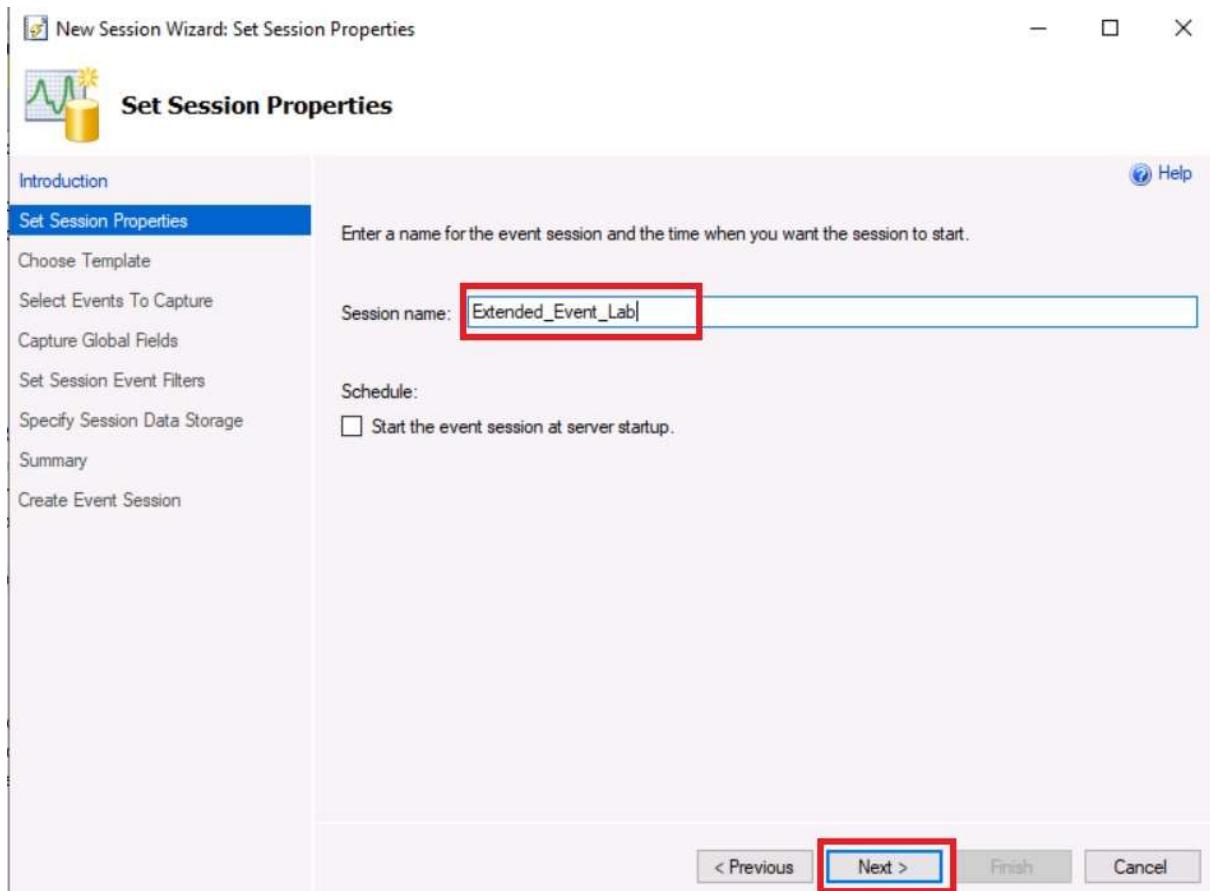
3. Click **New Session Wizard**

Expand **Databases - Salesdb - Extended Events** and right-click on **Sessions** and select **New Session Wizard**.



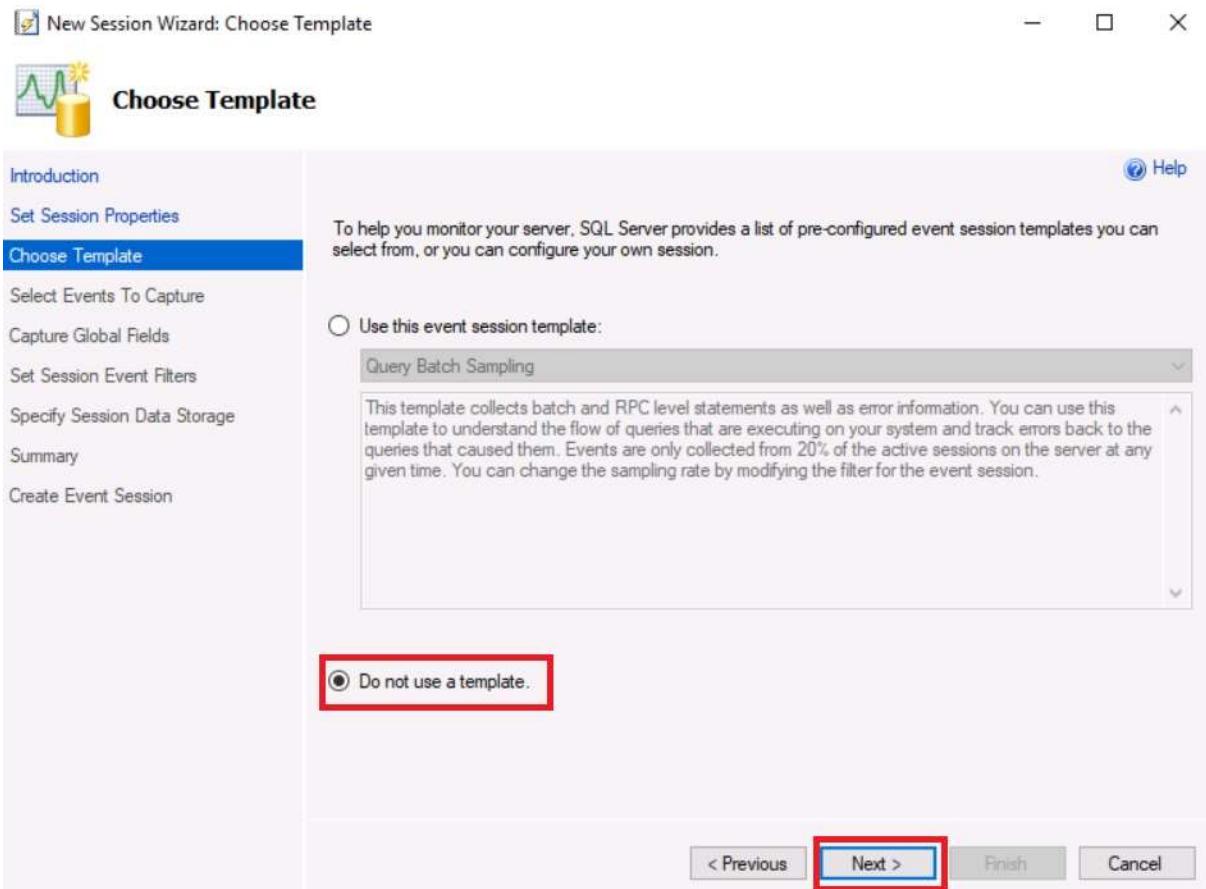
4. Configure **Set Session Properties**

Click Next at the Welcome page of the Wizard. (You may have to maximize the page to see the Next button). To configure "Set Session Properties", in **Session name:**, enter  **Extended\_Event\_Lab**. Click **Next**.



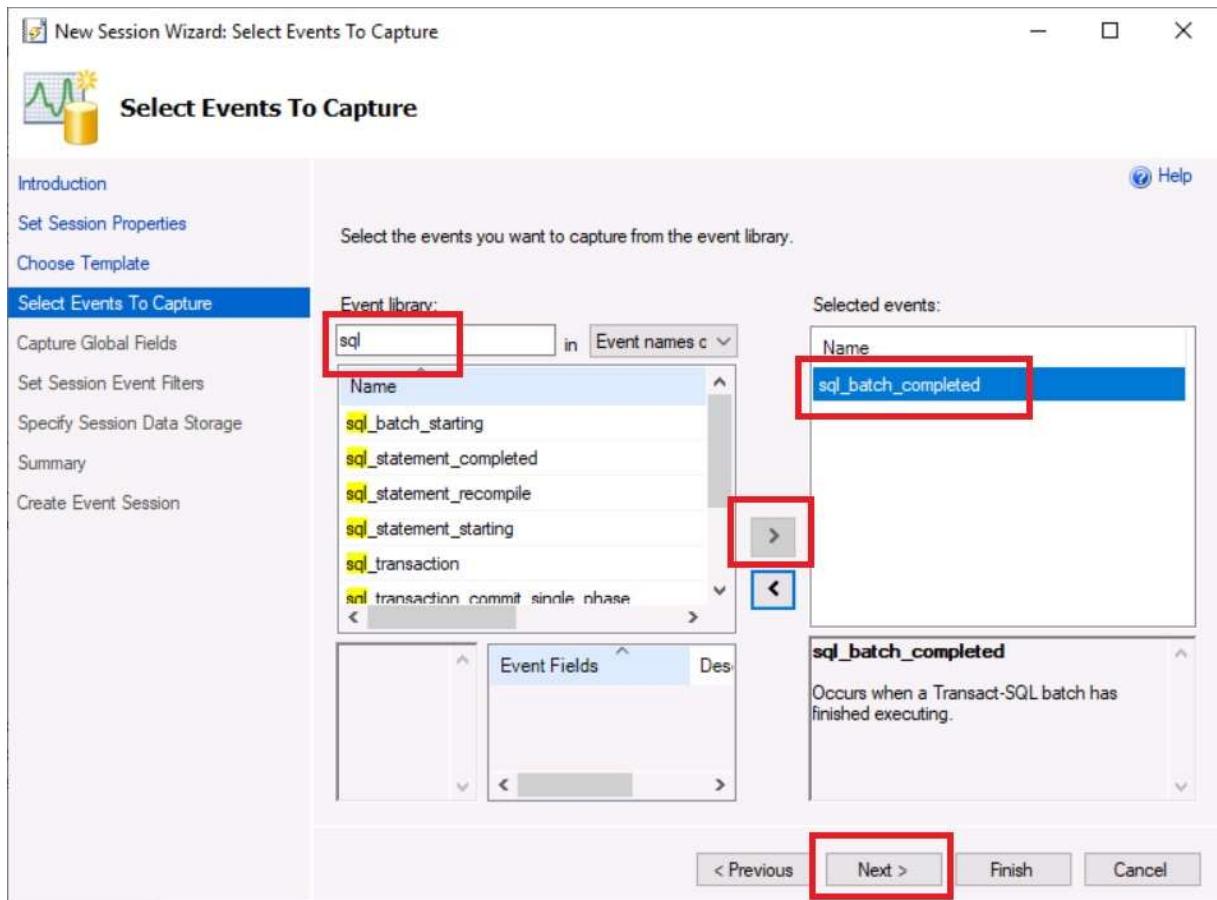
5. Configure **Choose template** page

Configure the **Choose Template** page. Keep all settings as **default**. ("Do not use a template"). Click **Next**.



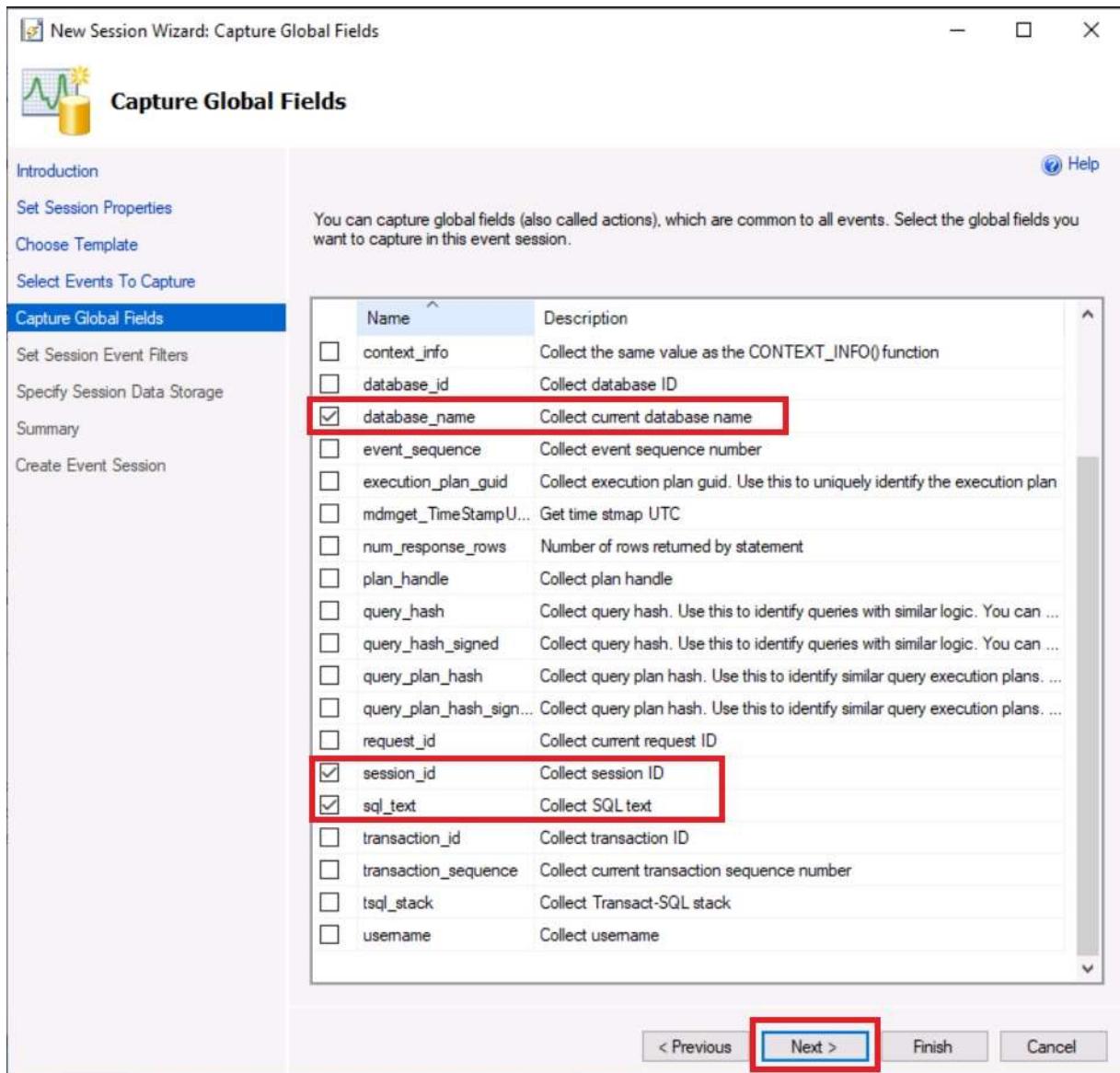
## □ 6. Configure **Select Events to Capture**

Configure **Select Events to Capture**. In **Selected events:**, select **sql\_batch\_completed**. Click the right-arrow icon to select the event. Click **Next**.



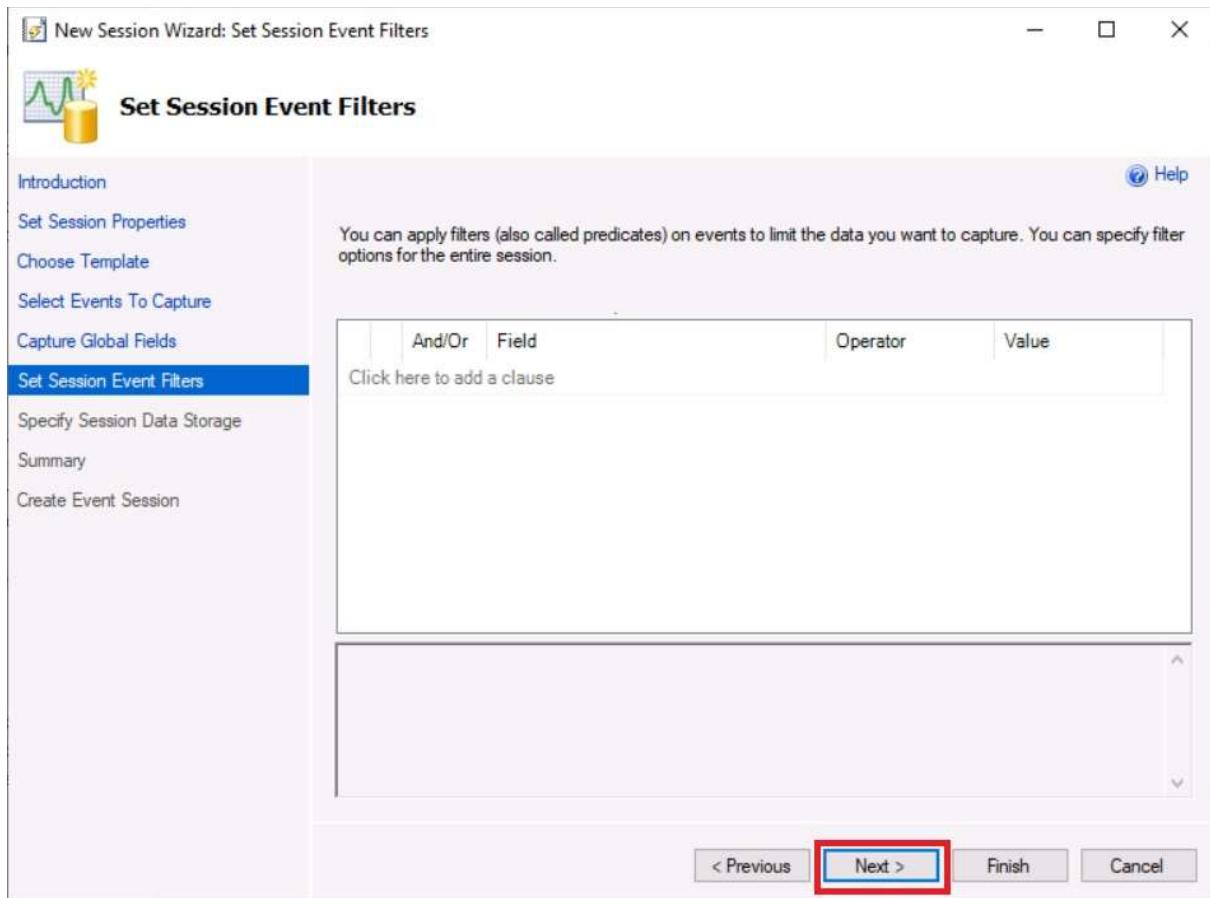
## 7. Configure **Capture Global Fields**

Configure **Capture Global Fields**. Check the **database\_name**, **session\_id**, **sql\_text** fields. Click **Next**.



#### □ 8. Configure **Set Session Event Filters**

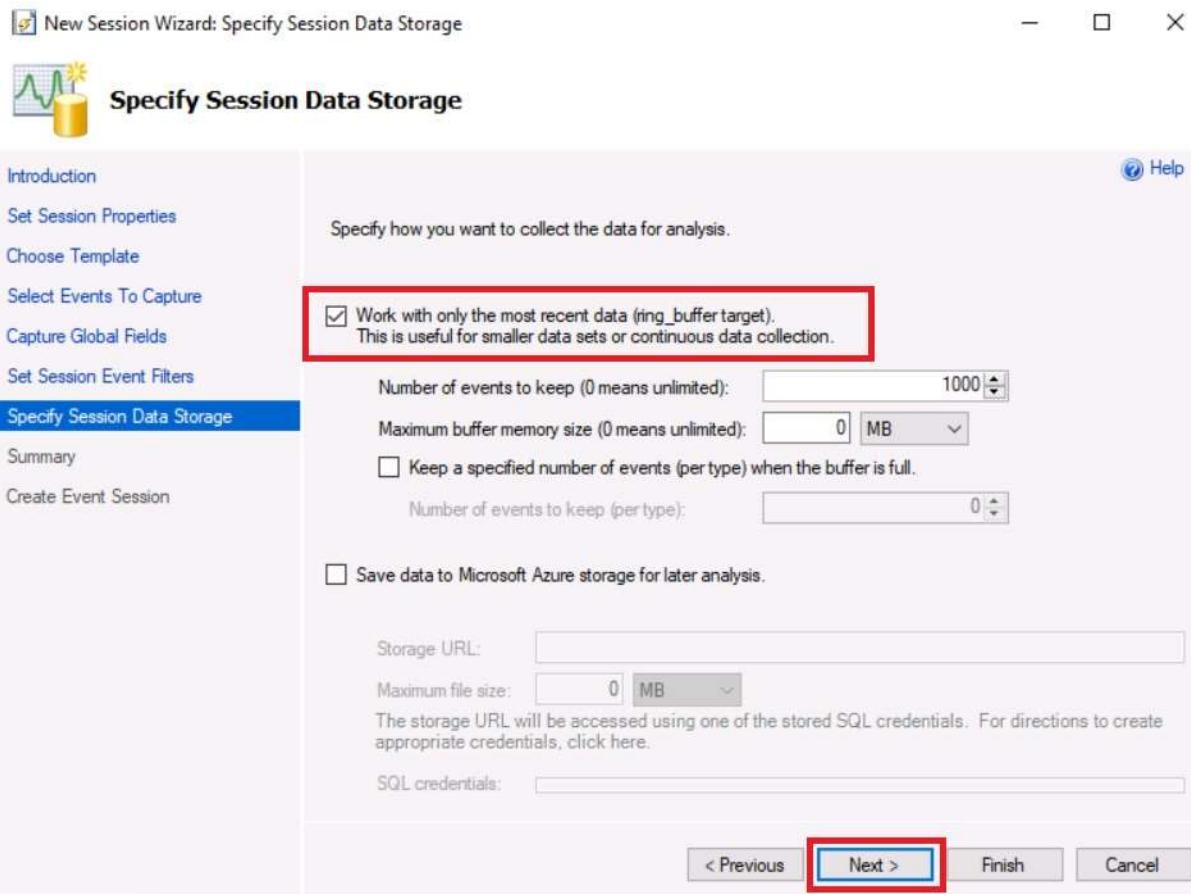
Configure **Set Session Event Filters**. Keep all settings as default. Click **Next**



#### 9. Configure **Specify Session Data Storage**

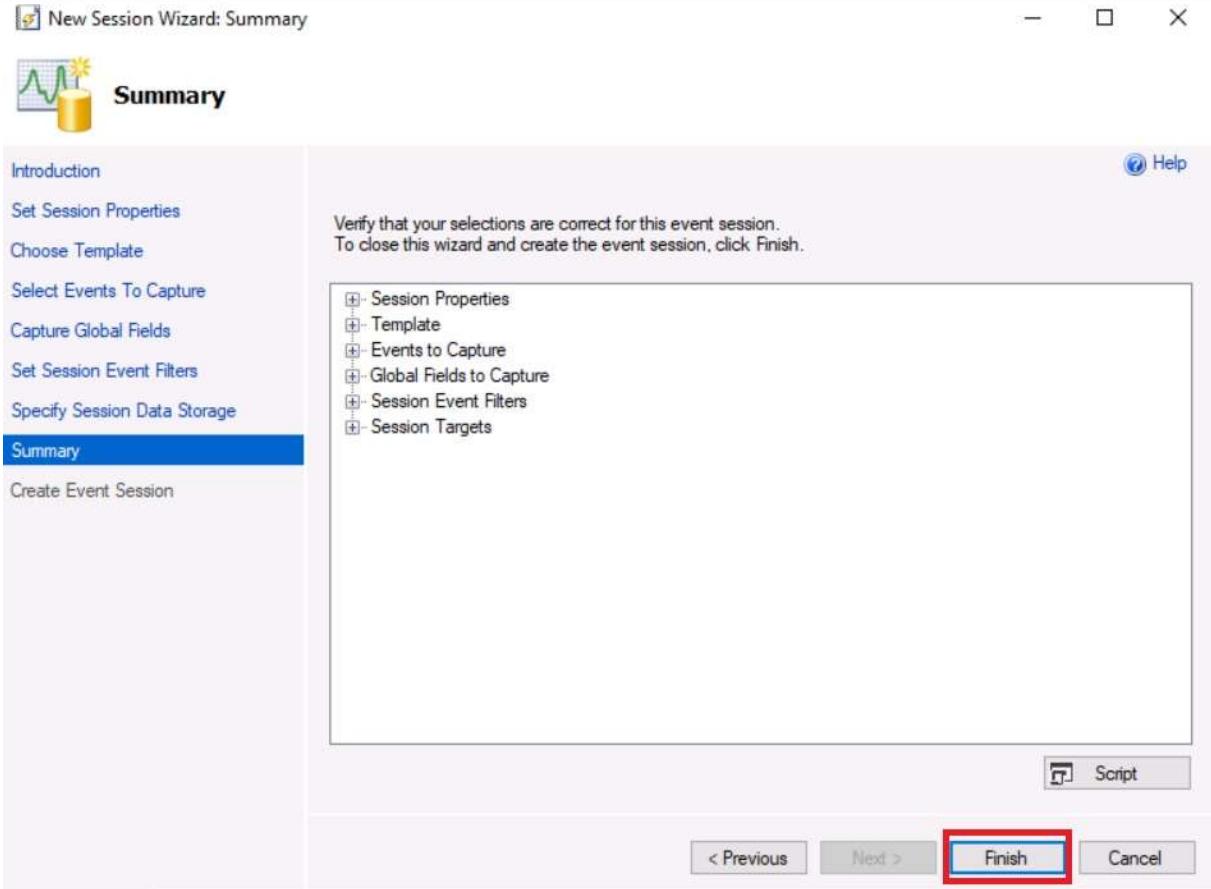
Configure **Specify Session Data Storage**. **Enable** Work with only the most recent data, and keep the rest as default. Click **Next**.

- i** For more information, please check this link - <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-xevent-db-diff-from-svr>



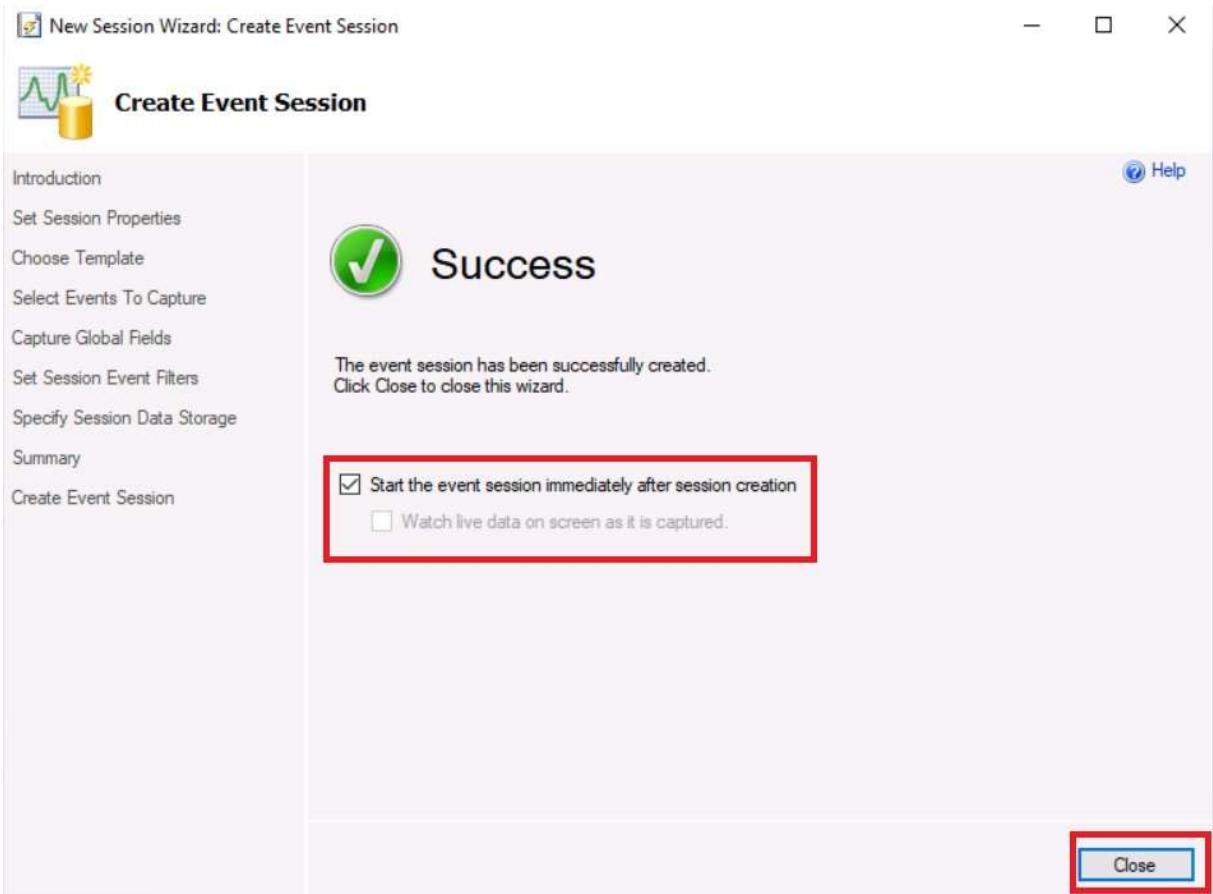
## 10. Review **Summary**

Review **Summary**. Click **Finish**



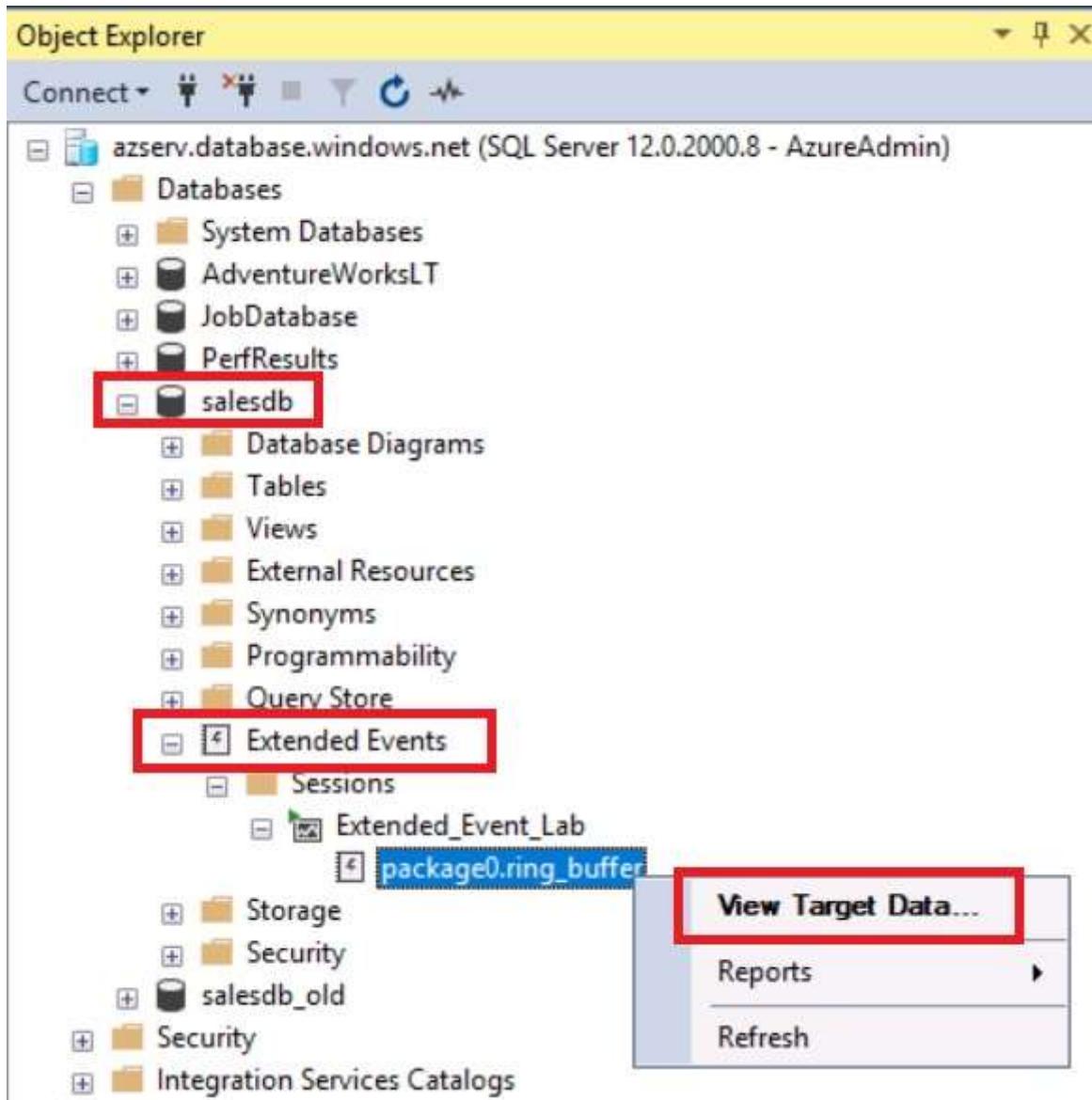
11. Start the Session

Enable **Start the event session immediately after session creation**. Click **Close**



12. Click View Target Data

Expand **Databases - salesdb - Extended Events - Sessions - Extended\_Event\_Lab** and Right-click on **package0.ring\_buffer** and Select **View Target Data**.



13. View XML output

Click the XML link on the right-hand side to view the extended events captured.

```
Extended_Event_Lab.ring_buffer1.xml  ✘ X salesdb - Extended...nt_Lab: ring_buffer
  ↴RingBufferTarget truncated="0" processingTime="0" totalEventsProcessed="11" eventCount="11" ↵
  ↴  <event name="sql_batch_completed" package="sqlserver" timestamp="2021-04-30T19:13:35.402Z">
  ↴    <data name="cpu_time">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴    <data name="duration">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>8165</value>
  ↴    </data>
  ↴    <data name="page_server_reads">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴    <data name="physical_reads">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴    <data name="logical_reads">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>4</value>
  ↴    </data>
  ↴    <data name="writes">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴    <data name="spills">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴    <data name="row_count">
  ↴      <type name="uint64" package="package0"></type>
  ↴      <value>0</value>
  ↴    </data>
  ↴  <!-- -->
```

Close all open windows without saving.

Congratulations!

You have successfully completed this lab. Click **Next** to advance to the next lab.

# Migrate a Compatible SQL Server Database to Azure SQL Database

---

## Introduction

In this lab, you will learn how to analyze your on-premises database, how to fix the compatibility issues and how to migrate your on-premises database to Azure SQL Database.

## Estimated Time

30 minutes

## Objectives

After completing this lab, you will be able to:

- Analyze your SQL Server Database for compatibility issues
- Fix database migration compatibility issues
- Migrate a Compatible SQL Server Database to Azure SQL Database

## Logon Information

1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as **[T @lab.VirtualMachine(AzureSQLDB).Username** using **[T @lab.VirtualMachine(AzureSQLDB).Password** as the password.

Click the **[T Type Text** icon to enter the associated text into the virtual machine.

2. Change the screen resolution if required.

You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Analyze your SQL Server Database for compatibility issues

---

This exercise is to practice how you can analyze if your database is compatible to migrate to Azure SQL Database. Some of the SQL Server features are not supported in Azure SQL Database.

---

## Tasks

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as `T @lab.VirtualMachine(AzureSQLDB).Username` using `T @lab.VirtualMachine(AzureSQLDB).Password` as the password.

- 2. Open Data Migration Assistant

Open the Data Migration Assistant (DMA) from the Windows Task Bar.

- 3. Click on the + button and create a new Assessment Project

Specify the following Project Properties and click on the **Create** Button

Property	Value
<b>Project Name</b>	<code>T AzureDBAssessement</code>
<b>Assessment Type</b>	Database Engine
<b>Source Server Type</b>	SQL Server
<b>Target Server Type</b>	Azure SQL Database

- 4. Select report type options

Select **Check database compatibility** and **Check Feature Parity** and click **Next**:

Select report type



Check database compatibility

Discover migration blocking issues and deprecated feature



Check feature parity

Discover unsupported or partially-supported features and t  
re-engineering.



Benefit from new features (coming soon...)

Discover new SQL Database features that are applicable to

□ 5. Connect to a server

Connect to your local SQL Server Instance. Specify your Server Name using  **localhost** and use **Windows Authentication**.

- a. Check **Encrypt Connection**
- b. Check **Trust Server Certificate**.
- c. Click the **Connect** button.

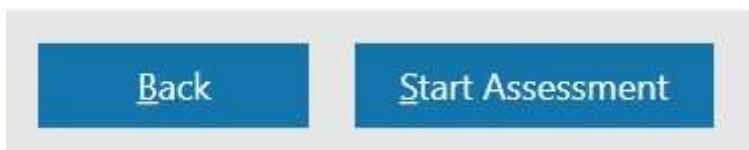


6. Select Sources

Select the database that you want to be part of the assessment. In this lab, select the **pubs** database. Leave the folder path empty and click **Add**.

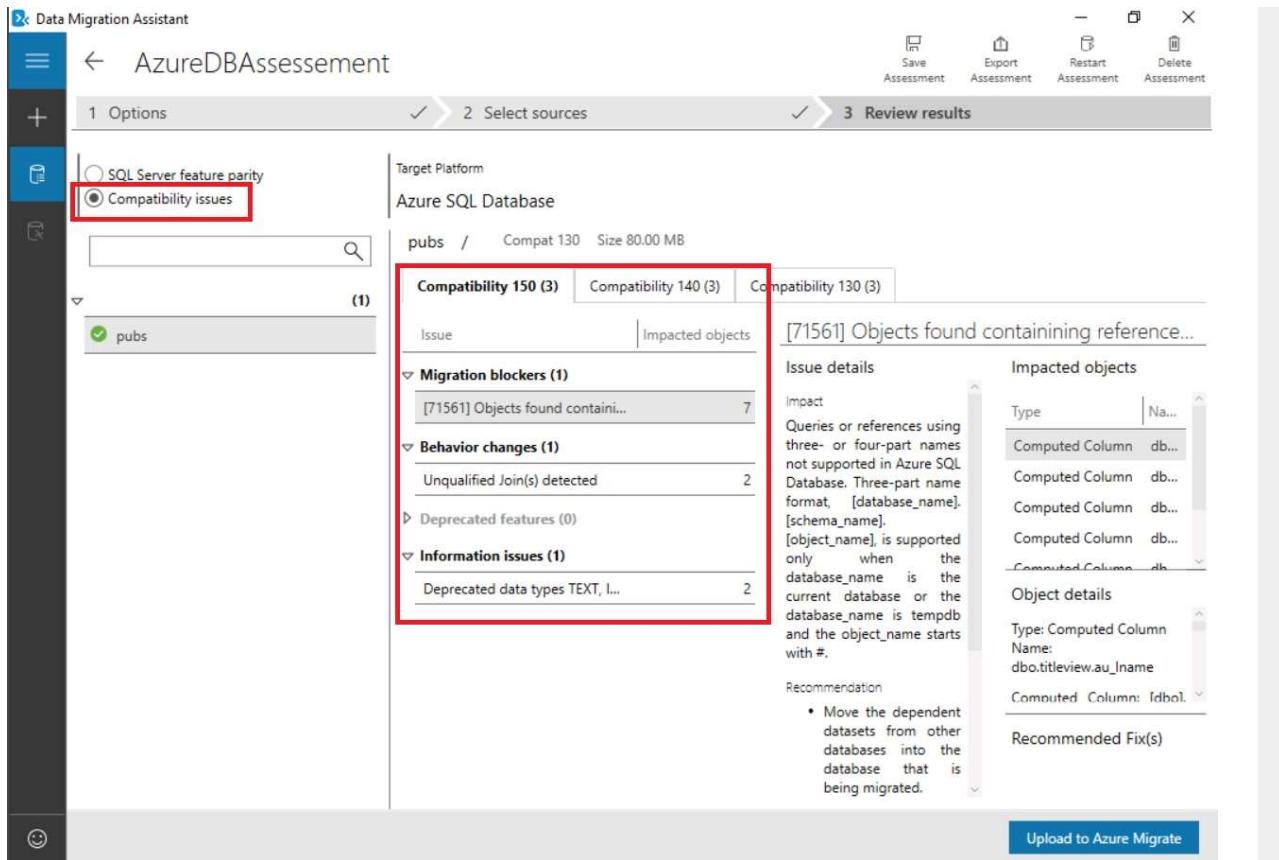
7. Start Assessment

Click the **Start Assessment** button and wait until the assessment is finished



8. Check the Compatibility Issues

Click on **Compatibility issues**, and wait until the assessment is finished.



Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Fix database migration compatibility issues

This exercise will help you to identify compatibility issues and how to solve them

### Tasks

- 1. Review the Migration Blocker

□ The issue is located in the view named **dbo.titleview**.  
It's using a Linked Server and four-part names which is not supported by Azure SQL Database.

[71561] Objects found containing references to unresolved objects,...

Issue details	Impacted objects
Queries or references using three- or four-part names not supported in Azure SQL Database. Three-part name format, [database_name].[schema_name]. [object_name], is supported only when the database_name is the current database or the database_name is tempdb and the object_name starts with #.	Computed Column dbo.titleview.au_ord Computed Column dbo.titleview.price Computed Column dbo.titleview.pub_id Computed Column dbo.titleview.title Computed Column dbo.titleview.ytd_s... View dbo.titleview

**Object details**  
Type: View  
Name: dbo.titleview  
View: [dbo].[titleview] contains an unresolved reference to an object. Either the

**Recommended Fix(s)**

- Move the dependent datasets from other databases into the database that is being migrated.
- Migrate the dependent database(s) to Azure and use 'Elastic Database Query' functionality to query across Azure SQL databases.

More info  
[Azure SQL Database elastic database query overview](#)

- 2. Update the view

Update the query in the view to make it compatible with Azure SQL Database.

Using SSMS, connect to the **T localhost** server using **Windows Authentication**. Expand the databases and open a new query window by right-clicking the **pubs** database and choosing **New Query** and change the view into:

SQL

```
▶ ALTER VIEW [dbo].[titleview] AS  
    select title, au_ord, au_lname, price, ytd_sales, pub_id  
    from authors  
    join titleauthor on authors.au_id = titleauthor.au_id  
    join titles on titles.title_id = titleauthor.title_id
```

**i** The script is located at **C:\LabFiles\M06L05Lab01** folder of the virtual machine.

### 3. Restart the Assessment

Go back to the DMA Assessment Project and click the button **Restart Assessment**. You will notice that the migration blocker is resolved, which mean we now have a compatible database that we can migrate to Azure SQL Database

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

# Exercise 3: Migrate a database to Azure with Data Migration Assistant

---

This exercise is to practice migration using **Data Migration Assistant**.

## Tasks

- 1. Connect to Microsoft Azure Portal

Open web browser and navigate to <http://portal.azure.com> to connect to Microsoft Azure Portal. Login as `[T] @lab.CloudPortalCredential(LabUser).Username` using `[T] @lab.CloudPortalCredential(LabUser).Password` as the password.

- 2. Create a new Azure Database

Go to your logical server <**LogicalServerName**> that you have previously created and create a new empty database with the following parameters:

Parameter	Value
<b>Subscription</b>	Leave default
<b>Resource Group</b>	Leave default

Parameter	Value
Database Name	<input type="text" value="Tpubs"/>
Server	<LogicalServerName> should be filled in by default
Want to use SQL Elastic Pool?	No
Compute + Storage	Click in Configure database and choose <b>Basic</b> . Click Apply
Backup storage redundancy	If this option shows to you, click <b>Geo-redundant backup storage</b> .
Collation	Go to Additional settings and choose <b>Collation Latin1_General_CI_AS</b> .

Click **Review + create** and then **Create**.

#### 3. Create a new Migration Project

Go to the DMA, click on the + button and create a new Migration Project by selecting the **Migration** radio button.

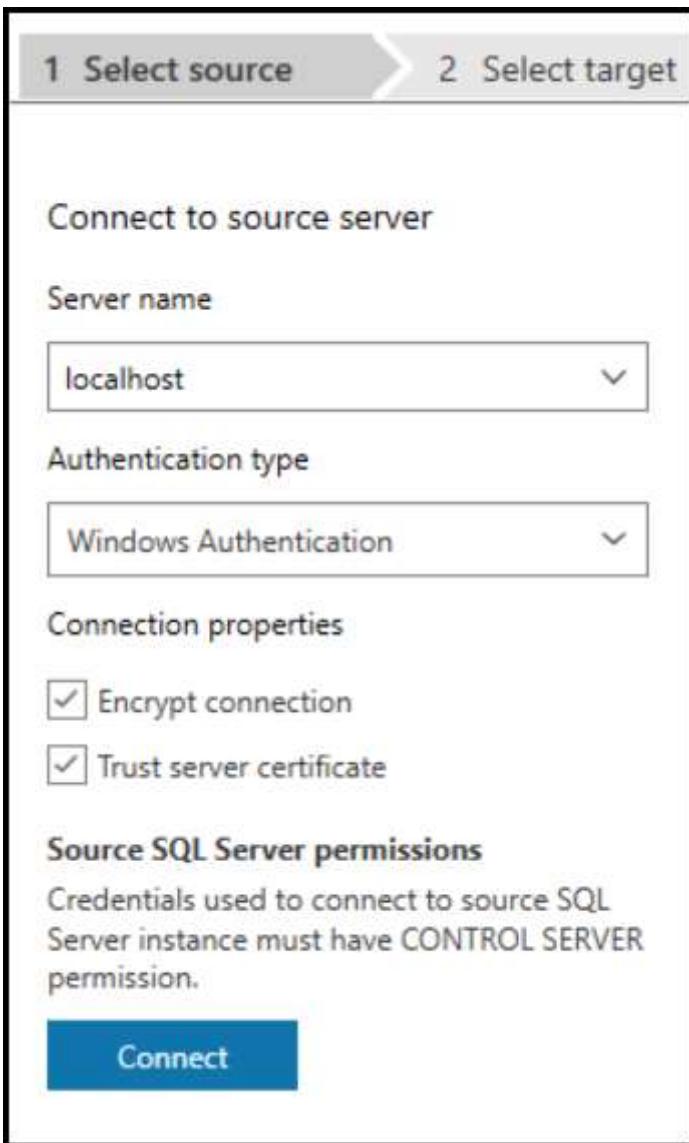
Specify the following Project Properties and click on the **Create** Button

Property	Value
Project Name	<input type="text" value="AzureDBMigration"/>
Source Server Type	SQL Server
Target Server Type	Azure SQL Database
Migration Scope	Schema and Data

#### 4. Select to a Source Server

Connect to your local SQL Server Instance by specifying the Server Name as  and use **Windows Authentication**.

- a. Check **Encrypt Connection**
- b. Check **Trust Server Certificate**.
- c. Click the **Connect** button.



□ 5. Select the database

Select the database from the source that you want to migrate. In this lab, select the **pubs** database.

**Uncheck the "Assess database before migration"** because you already did the assessment previously. Click **Next**.

Name	Compatibility Level	Assess database before migration?
pubs	130	<input checked="" type="checkbox"/>

□ 6. Select the target server

Specify the following settings to connect to your target server and click **Connect**.

Setting	Value
Server name	<input type="text"/> <LogicalServerName>.database.windows.net
Authentication Type	SQL Authentication
Username	<input type="text"/> AzureAdmin
Password	<input type="text"/> Passw0rd1

**Check "Encryption connection" and "Trust server certificate" box. Click Connect.**

- 7. Select the target database

Select the **pubs** database. This is the empty database that you have created earlier. Click **Next**.

AzureDBMigration

1 Select source ✓ 2 Select target 3 Select objects 4 Script & deploy schema 5 Select tables 6 Migrate data

Create a new Azure SQL Database...

Server name: azserv.database.windows.net

Authentication type: SQL Server Authentication

SQL Authentication credentials:

Username: AzureAdmin

Password: \*\*\*

Connection properties:

Encrypt connection

Trust server certificate

Target Azure SQL Database permissions:

The principal used to connect must have CONTROL DATABASE permission on the target database.

Connect

Back Next

- 8. Select Objects

Select all the objects because we want to migrate the complete database.

- 9. Generate Script

Click the **Generate SQL script** button.

- 10. Deploy the schema

Once the script is generated, click the button **Deploy Schema** to deploy the database schema in your Azure SQL Database named **pubs**.

□ 11. Migrate the data

**i** After the schema is successfully deployed, we need to fill the tables with data.

Click the **Migrate data** button.

□ 12. Select Tables

Select all the tables that contain data to be migrated and click the **Start data migration** button

AzureDBMigration

Delete Migration

1 Select source ✓ > 2 Select target ✓ > 3 Select objects ✓ > 4 Script & deploy ✓ > 5 Select tables > 6 Migrate data

Source database	Target database
pubs localhost	pubs azserv.database.windows.net

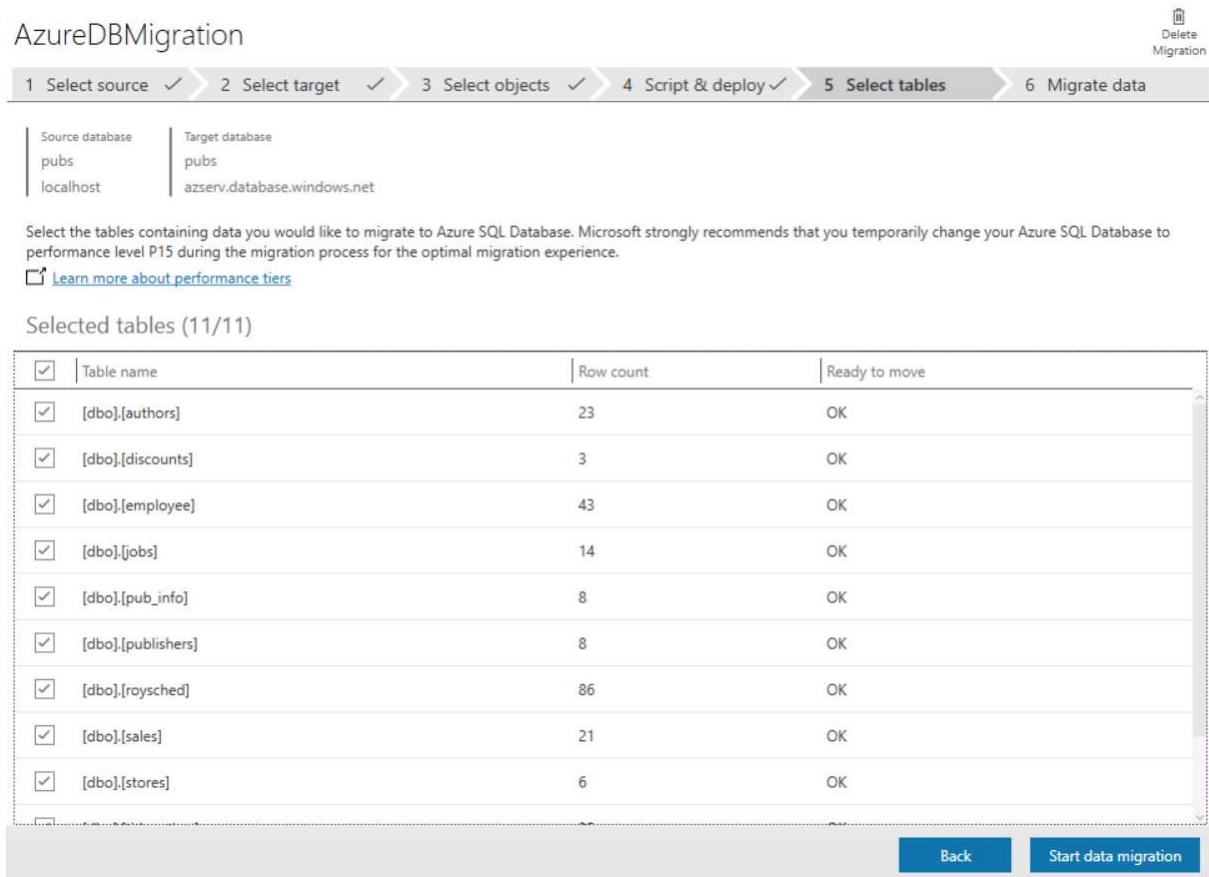
Select the tables containing data you would like to migrate to Azure SQL Database. Microsoft strongly recommends that you temporarily change your Azure SQL Database to performance level P15 during the migration process for the optimal migration experience.

[Learn more about performance tiers](#)

Selected tables (11/11)

<input checked="" type="checkbox"/>	Table name	Row count	Ready to move
<input checked="" type="checkbox"/>	[dbo].[authors]	23	OK
<input checked="" type="checkbox"/>	[dbo].[discounts]	3	OK
<input checked="" type="checkbox"/>	[dbo].[employee]	43	OK
<input checked="" type="checkbox"/>	[dbo].[jobs]	14	OK
<input checked="" type="checkbox"/>	[dbo].[pub_info]	8	OK
<input checked="" type="checkbox"/>	[dbo].[publishers]	8	OK
<input checked="" type="checkbox"/>	[dbo].[roysched]	86	OK
<input checked="" type="checkbox"/>	[dbo].[sales]	21	OK
<input checked="" type="checkbox"/>	[dbo].[stores]	6	OK

Back Start data migration



□ 13. Migration Complete

Verify that the migration completed successfully.

 11  
Server objects

 0  
In-progress

 11  
Successful

 0  
Warnings

 0  
Failed

## ▽ Tables (11)

Status	Table name	Migration details
	[dbo].[authors]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[discounts]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[employee]	Migration successful. Duration: 0 hrs 0 mins 2 secs
	[dbo].[jobs]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[pub_info]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[publishers]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[roysched]	Migration successful. Duration: 0 hrs 0 mins 2 secs
	[dbo].[sales]	Migration successful. Duration: 0 hrs 0 mins 2 secs
	[dbo].[stores]	Migration successful. Duration: 0 hrs 0 mins 3 secs
	[dbo].[titleauthor]	Migration successful. Duration: 0 hrs 0 mins 2 secs
	[dbo].[titles]	Migration successful. Duration: 0 hrs 0 mins 2 secs

14. Close all open windows without saving.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 4: Migrate a database to Azure with SSMS

---

This exercise is to practice migrating database using **SSMS**.

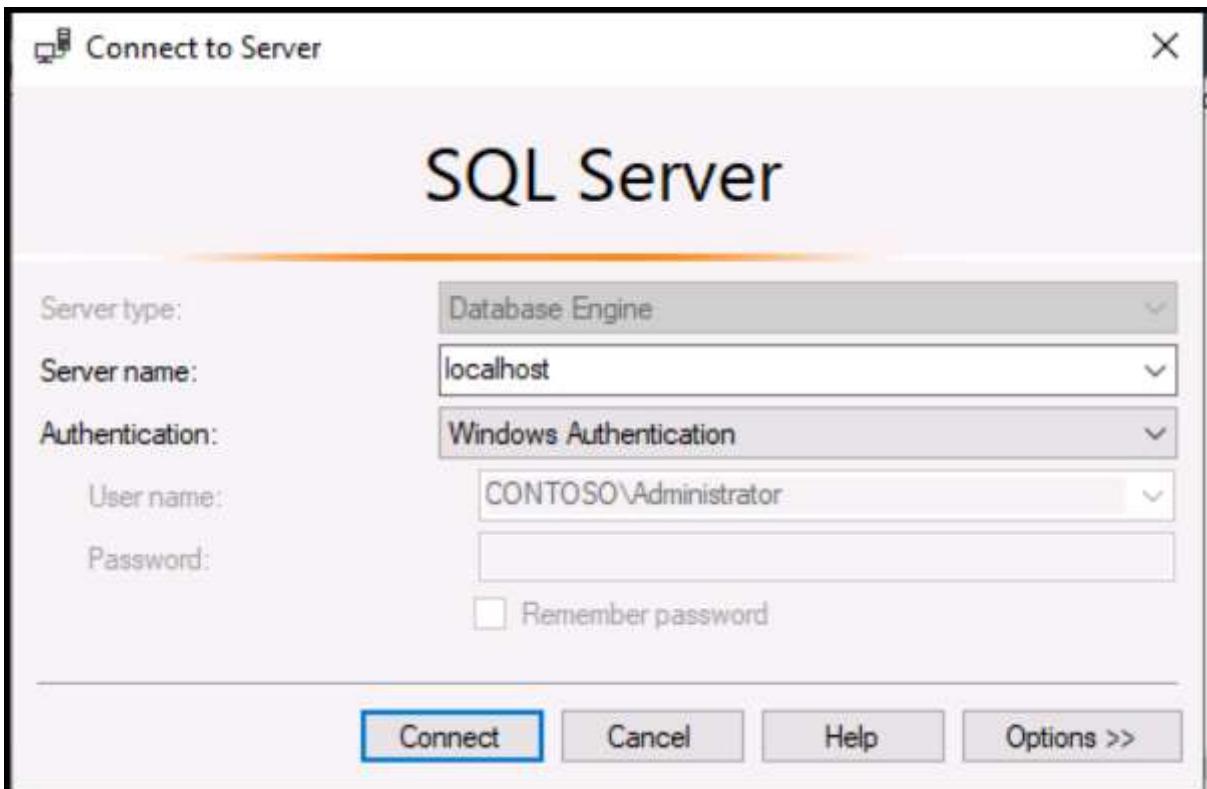
### Tasks

- 1. Connect to Microsoft Azure Portal

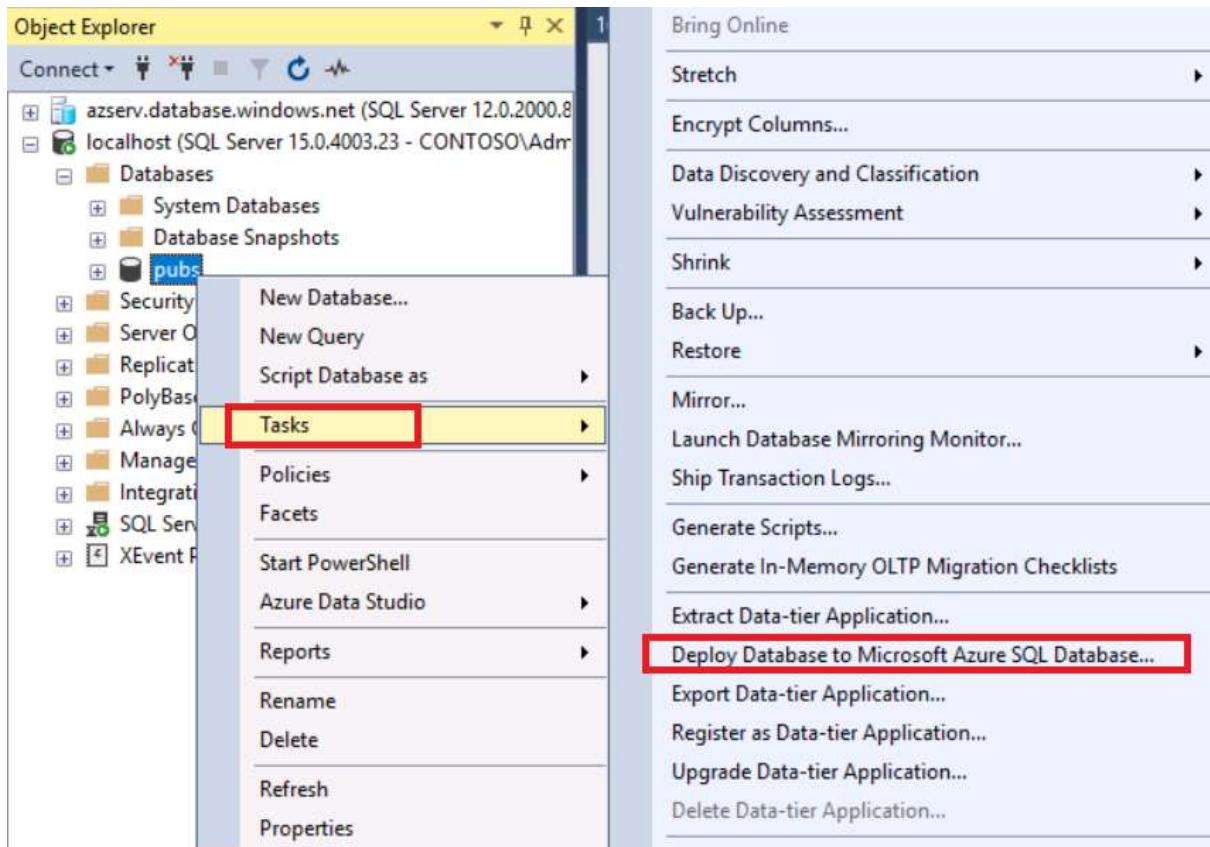
Open Internet Explorer and navigate to <http://portal.azure.com> to connect to Microsoft Azure Portal. Login as `@lab.CloudPortalCredential(LabUser).Username` using `@lab.CloudPortalCredential(LabUser).Password` as the password.

- 2. Select SQL Source

Connect to your local SQL Server Instance using **SSMS** by specifying the Server Name as `localhost` and use **Windows Authentication**.



Browse to **pubs** database. Right-Click and select **Tasks -> Deploy Database to Microsoft Azure SQL Database**



3. Select the target Server

On the **Deploy Database to Microsoft Azure SQL Database** Click **Next>**.

On Server connection, click on **connect** and enter information below to connect to Azure SQL Instance

Specify the following settings to connect to your target server and click **Connect**.

Setting	Value
Server name	<input type="text"/> <LogicalServerName>.database.windows.net
Authentication Type	SQL Authentication
Username	<input type="text"/> AzureAdmin
Password	<input type="text"/> Passw0rd1

### Specify Target Connection

Specify the name of the instance of SQL Server or the Microsoft Azure SQL Database server that will host the deployed database, name the new database, and then click Connect to login to the target server.

Server connection:

azserv (AzureAdmin)

Connect...

New database name:

pubs

Microsoft Azure SQL

Edition of Microsoft A

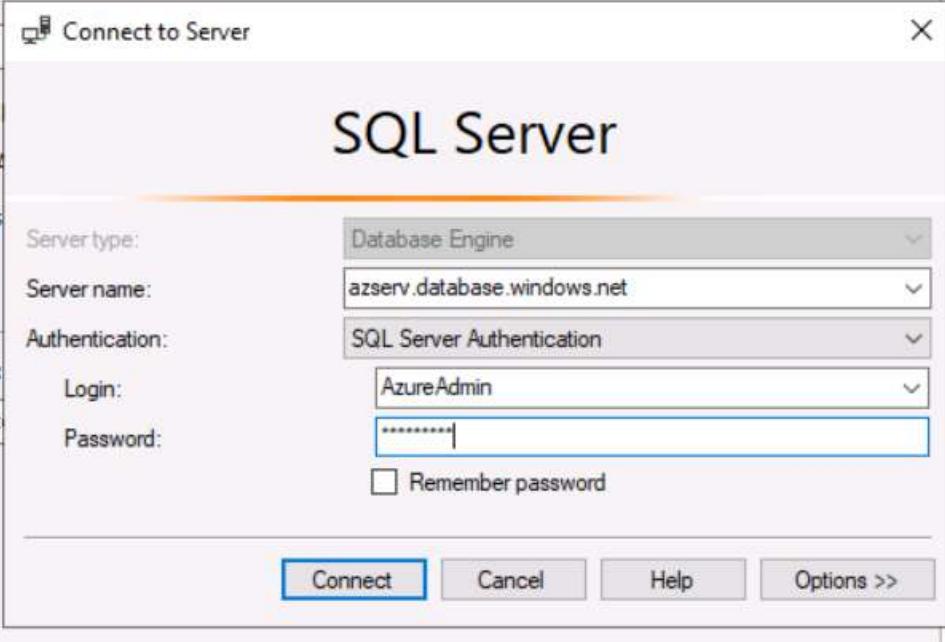
Maximum database s

Service Objective :

Other settings

Temporary file name:

C:\Users\Administrat



Specify the following Microsoft Azure SQL Database settings and click on the **Next** Button

Property	Value
New Database Name	T ssmspuds
Edition	Standard
Maximum DB Size	10
Service Objective	S0

### Specify Target Connection

Specify the name of the instance of SQL Server or the Microsoft Azure SQL Database server that will host the deployed database, name the new database, and then click Connect to login to the target server.

Server connection:

New database name:

Microsoft Azure SQL Database settings

Edition of Microsoft Azure SQL Database:

Maximum database size (GB):

Service Objective :

Other settings

Temporary file name:

Click **Finish** on summary.

- 4. Migration Complete

Validate successful migration and click **Close**

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

# Exercise 5: Migrate a database to Azure with Transactional Replication

---

This exercise is to practice migration using Transactional Replication

---

## Tasks

- 1. Connect to Microsoft Azure Portal

Open Internet Explorer and navigate to [T http://portal.azure.com](http://portal.azure.com) to connect to Microsoft Azure Portal. Login as [T @lab.CloudPortalCredential\(LabUser\).Username](#) using [T @lab.CloudPortalCredential\(LabUser\).Password](#) as the password.

- 2. Create a new database

Go to your logical server <**LogicalServerName**> that you have previously created and create a new empty database with the following parameters:

Parameter	Value
<b>Subscription</b>	Leave it default
<b>Resource group</b>	Leave it default
<b>Database Name</b>	<a href="#">T pubsrepl</a>
<b>Server</b>	< <b>LogicalServerName</b> > should be filled in by default
<b>Want to use SQL Elastic Pool?</b>	No
<b>Compute + Storage</b>	Click in Configure database and choose <b>Standard</b> . Click Apply
<b>Backup storage redundancy</b>	If this option shows to you, select <b>Geo-redundant backup storage</b>

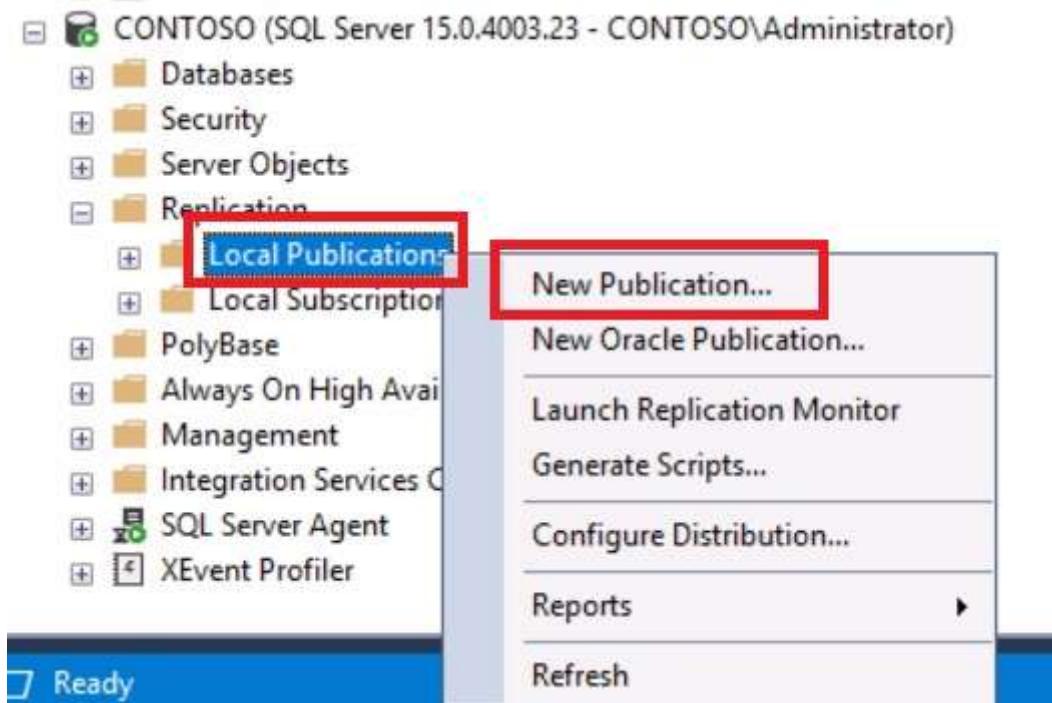
Click **Review + create** and then **Create**.

- 3. Configure New Publication

Connect to your local SQL Server Instance using **SSMS** by specifying the Server Name as [T CONTOSO](#) and use **Windows Authentication**. If **SQL Server Agent** is not started, right-click it and choosing **Start**.

**i** This service will be needed by Replication.

Expand **Replication**, right-click on **Local Publications** and select **New Publication**. Click **Next**.



4. Configure Distributor

Keep your local on-premises server as Distributor (leave it as default). Click **Next**.

5. Configure Snapshot Folder

Keep the setting as default. Click **Next**.

6. Configure Publication Database

Select **pubs** database. Click **Next**.

7. Configure Publication Type

Select **Transactional Publication**. This is the only publication type that is supported to Azure SQL Database. Click **Next**.

8. Configure Articles

Select all the objects (tables, views, stored procedures) in the Articles windows and click **Next**.

9. Configure Article Issues

Click **Next** in the Article Issues windows. These are not actual issues but just informational messages for this database.

10. Configure Filter Table Rows

We are not going to filter any rows from the tables, so just click **Next** in the Filter Table Rows window.

11. Configure Snapshot Agent

In the Snapshot Agent window, select the checkbox "Create the snapshot immediately and keep the snapshot available to initialize subscriptions". Click **Next**.

12. Configure Agent Security

Click the **Security Settings** button in the Agent Security window. Select "Run under the following Windows Account" and specify **T** **CONTOSO\Administrator** as the process account and **T** **Pa\$\$wOrd** as the password.

Click **OK**. Click **Next**.

13. Configure Wizard Actions

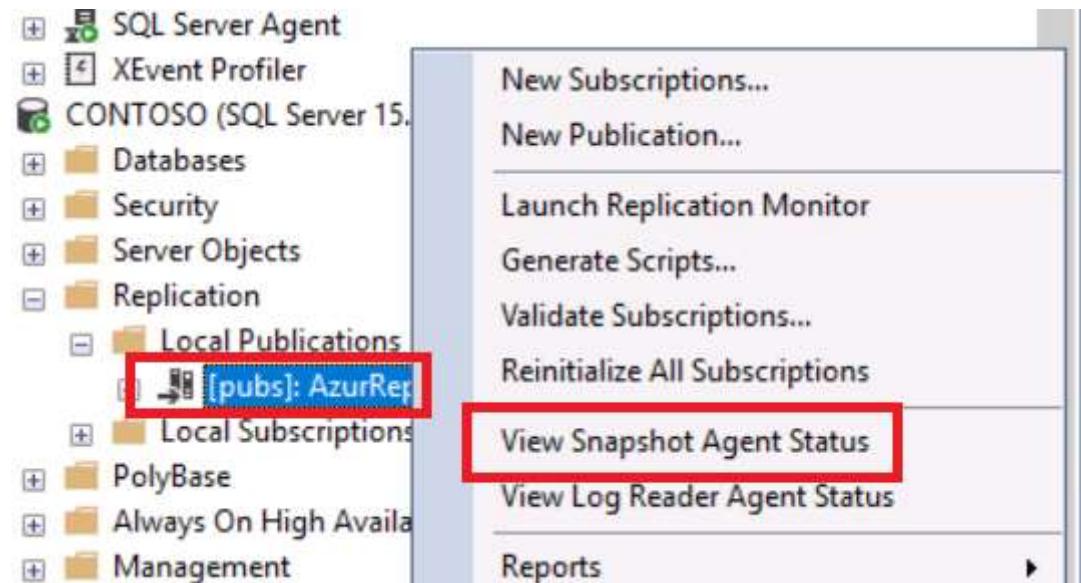
Select "Create the publication" and Click **Next**.

14. Configure Complete the Wizard

Set **publication name** as **T** **AzurRepl** and Click **Finish**. Click **Close**.

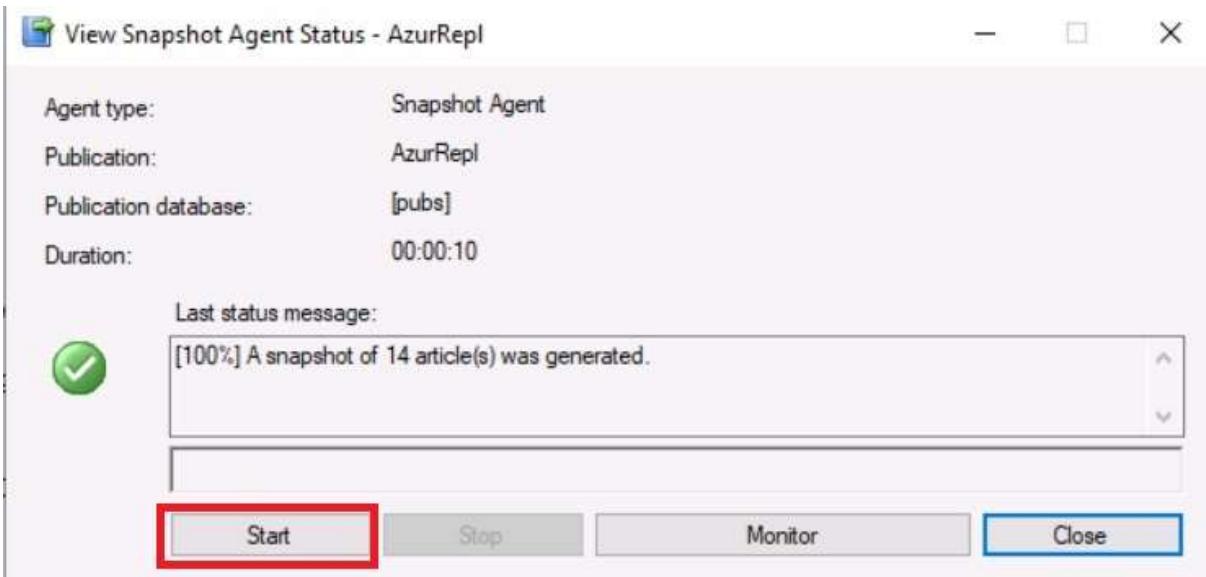
15. Select **View Snapshot Agent Status**

Right Click on AzurRepl publication and Select View Snapshot Agent Status.



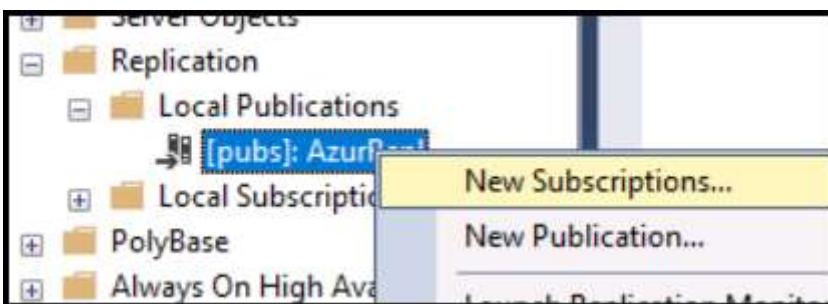
16. View Snapshot Agent

Right-click the new publication and choose **View Snapshot Agent Status**. If it is not started, click **Start**.



17. Select New Subscriptions

Right click on the AzurRepl publication and select New Subscription.



18. Configure Publication

Select **AzurRepl** as a publication and click **Next**.

19. Configure Distribution Agent Location

Select the option "**Run all agents at the Distributor CONTOSO (push subscriptions)**".

**i** This is the only option that is supported by Azure SQL Database.

Click **Next**.

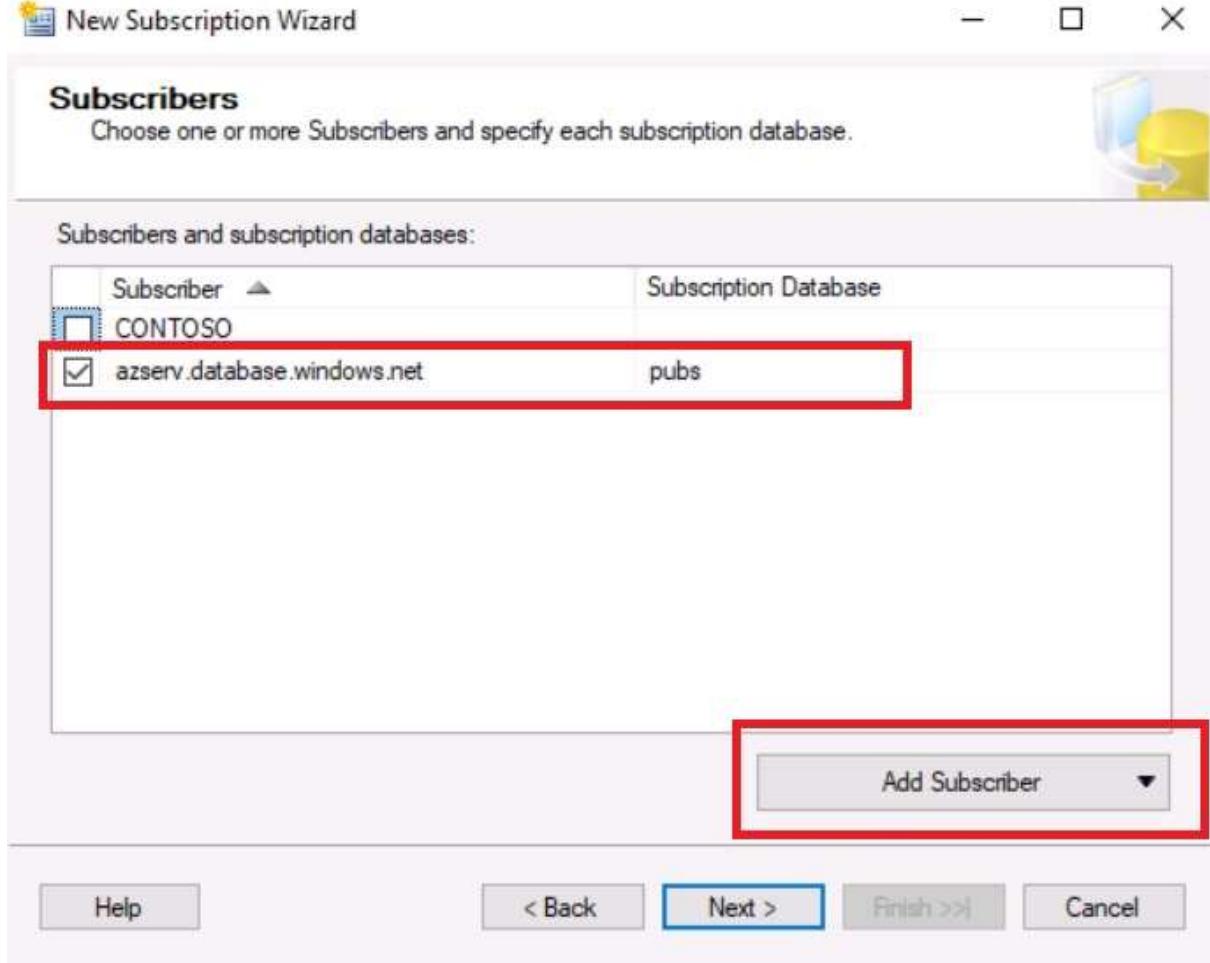
20. Configure Subscribers

Click **Add Subscriber** button and select **Add SQL Server Subscriber**. Connect to the logical server <LogicalServerName>.database.windows.net using the following information.

Setting	Value
Server name	<input type="text" value="T &lt;LogicalServerName&gt;.database.windows.net"/>

Setting	Value
<b>Authentication</b>	SQL Server Authentication
<b>User name</b>	<input type="text"/> AzureAdmin
<b>Password</b>	<input type="text"/> Passw0rd1

After the connection select **pubsrep1** database as a **Subscription Database**. Click **Next**.



#### 21. Configure Distribution Agent Security

In the **Distribution Agent Security Window** click in the [...] button. Configure the following detailed parameters.

Parameter	Value
<b>Specify the domain or machine account</b>	Select Run under the following Windows Account
<b>Process Account</b>	<input type="text"/> CONTOSO\Administrator

Parameter	Value
<b>Password</b>	<input type="text"/> Pa\$\$w0rd
<b>Connect to Distributor</b>	Select By impersonating the process account
<b>Connect to Subscriber</b>	Select Using the following SQL Server login
<b>Login</b>	<input type="text"/> AzureAdmin
<b>Password</b>	<input type="text"/> Passw0rd1
<b>Confirm Password</b>	<input type="text"/> Passw0rd1

Click **OK**. Click **Next**.

22. Configure Synchronization Schedule

Select "Run continuously" as the Agent Schedule in the Synchronization Schedule windows. Click **Next**.

23. Configure Initialize Subscriptions

Select "Immediately" as the Initialize When option in the Initialize Subscriptions windows, click **Next**.

24. Configure Wizard Actions

Select the Create the subscription(s) option in the Wizard Actions window and click **Next**.

25. Complete the Wizard

Click **Finish** in the Complete the Wizard windows and **Wait** until the subscription is created. Click **Close**.

26. Select Launch Replication

Go to the object explorer and right click on the **AzurRepl publication** and select "**Launch Replication Monitor**"

27. Click on tab Subscription Watch List

Set Show to **All Subscriptions** and double click on the **Subscription**.

Status	Subscription	Publication	Performance	Latency	Last Synchronization
Running	[AZSERV.DATAB...]	[pubs]: AzurRepl	Excellent	00:00:00	5/7/2021 8:35:33 ...

28. Verify the replication

Check all **Bulk copied** messages.

The screenshot shows the 'Subscription AZSERV.DATABASE.WINDOWS.NET:pubs to CONTOSO:pubs:AzurRepl' window. The 'Publisher To Distributor History' tab is selected. The 'View:' dropdown is set to 'The last 100 synchronizations'. The 'Sessions of the Distribution Agent' table shows one session: 'Running' at 5/7/2021 8:33:55 AM, duration 00:02:39. The 'Actions in the selected session:' table lists various replication events. A red box highlights the 'Bulk copied data into table' entries, which include: 'Bulk copied data into table 'titleauthor' (25 rows)', 'Bulk copying data into table 'titleauthor'', 'Bulk copied data into table 'sales' (21 rows)', 'Bulk copying data into table 'sales'', 'Bulk copied data into table 'titles' (18 rows)', 'Bulk copying data into table 'titles'', 'Bulk copied data into table 'stores' (6 rows)', 'Bulk copying data into table 'stores'', 'Bulk copied data into table 'pub\_info' (8 rows)', and 'Bulk copying data into table 'pub\_info''. The 'Action Time' column for these events ranges from 5/7/2021 8:34:27 AM to 5/7/2021 8:36:34 AM.

Action Message	Action Time
No replicated transactions are available.	5/7/2021 8:36:34 AM
Delivered snapshot from the 'unc\CONTOSO_PUBS_AZURREPL\20210507082752\' su...	5/7/2021 8:34:29 AM
Creating Primary Key index on table 'dbo1.temployee'	5/7/2021 8:34:28 AM
<b>Bulk copied data into table 'titleauthor' (25 rows)</b>	5/7/2021 8:34:28 AM
Bulk copying data into table 'titleauthor'	5/7/2021 8:34:28 AM
Bulk copied data into table 'sales' (21 rows)	5/7/2021 8:34:28 AM
Bulk copying data into table 'sales'	5/7/2021 8:34:27 AM
Bulk copied data into table 'titles' (18 rows)	5/7/2021 8:34:27 AM
Bulk copying data into table 'titles'	5/7/2021 8:34:27 AM
Bulk copied data into table 'stores' (6 rows)	5/7/2021 8:34:27 AM
Bulk copying data into table 'stores'	5/7/2021 8:34:27 AM
Bulk copied data into table 'pub_info' (8 rows)	5/7/2021 8:34:27 AM
Bulk copying data into table 'pub_info'	5/7/2021 8:34:27 AM

29. Verify the replication through **SSMS**

Connect with the SSMS to the **pubsrepI** database and verify if all objects and data have been migrated.

The screenshot shows the SQL Server Object Explorer interface. The connection is to 'azserv.database.windows.net (SQL Server 12.0.2000.8 - AzureAdmin)'. Under the 'Databases' node, several system databases are listed: System Databases, AdventureWorksLT, JobDatabase, PerfResults, pubs, and pubsrepl. Below the databases is a 'Database Diagrams' node. A red box highlights the 'Tables' node under 'Database Diagrams', which is expanded to show 12 system tables: System Tables, External Tables, GraphTables, dbo.authors, dbo.discounts, dbo.employee, dbo.jobs, dbo.pub\_info, dbo.publishers, dbo.roysched, dbo.sales, dbo.stores, dbo.titleauthor, and dbo.titles. Other nodes shown include Views, External Resources, Synonyms, Programmability, and Query Store.

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next lab.

# Azure SQL Data Sync

---

## Introduction

In this lab, you will learn how to setup Azure SQL Data Sync between 2 databases on the same logical server and one database on a second logical server.

## Estimated Time

20 minutes

## Objectives

After completing this lab, you will be able to:

- Set up Azure SQL Data Sync

## Logon Information

- 1. Connect to `@lab.VirtualMachine(AzureSQLDB).SelectLink` as  `@lab.VirtualMachine(AzureSQLDB).Username` using  `@lab.VirtualMachine(AzureSQLDB).Password` as the password.
  - Click the  **Type Text** icon to enter the associated text into the virtual machine.
- 2. Change the screen resolution if required.
  - You may want to adjust the screen resolution to your own preference. Do this by right-clicking on the desktop and choosing **Screen resolution** and clicking **OK** when finished.

[Back to the list of Labs](#)

# Exercise 1: Create a new Sync Group

---

This exercise shows how to create a new Sync Group

---

## Tasks

- 1. Connect to **@lab.VirtualMachine(AzureSQLDB).SelectLink** as **T @lab.VirtualMachine(AzureSQLDB).Username** using **T @lab.VirtualMachine(AzureSQLDB).Password** as the password.
- 2. Connect to the Azure Portal and browse to your SQL Server (logical server) **<LogicalServerName>**.

Open the Azure Portal in your browser **T https://portal.azure.com/**

Login as **T @lab.CloudPortalCredential(LabUser).Username** using **T @lab.CloudPortalCredential(LabUser).Password** as the password.

- 3. Create a new empty database on your logical server

Create a new empty database with the following settings

Parameter	Value
<b>Subscription</b>	Leave it default
<b>Resource group</b>	Leave it default
<b>Database Name</b>	<b>T salesdbSync</b>
<b>Server</b>	<b>&lt;LogicalServerName&gt;</b> should be filled in by default
<b>Want to use SQL Elastic Pool?</b>	No
<b>Compute + Storage</b>	Click in Configure database and choose <b>Standard</b> . Click <b>Apply</b>
<b>Backup storage redundancy</b>	If this option shows to you, click <b>Geo-redundant backup storage</b> .

Click **Review + create** and then **Create**.

- 4. Create a new empty database on your Disaster Recover logical server

Create a new empty database on your Disaster Recovery logical server which you have created in a previous lab of Module 3. Use the following settings

Setting	Value
<b>Subscription</b>	Leave it default
<b>Resource group</b>	Leave it default
<b>Database Name</b>	<input type="text" value="salesdbSync"/>
<b>Server</b>	<LogicalServerName> should be filled in by default
<b>Want to use SQL Elastic Pool?</b>	No
<b>Compute + Storage</b>	Click in Configure database and choose <b>Standard</b> . Click Apply
<b>Backup storage redundancy</b>	If this option shows to you, click <b>Geo-redundant backup storage</b> .

5. Configure a new Sync Group

Go to your **Salesdb** that you have created in a previous lab

- If you have completed a Failover Group in a previous lab, make sure you select the **Salesdb** database from the current primary logical SQL server.

6. Under **Data management** select **Sync to other databases** option and then click on the **New Sync group** button.

The screenshot shows the Azure portal interface for managing databases. On the left, there's a sidebar with various options like Compute + storage, Connection strings, Properties, Locks, Data management, Replicas, and Sync to other databases. The 'Sync to other databases' option is highlighted with a red box. The main area is titled 'Sync Group' and shows a message: 'You do not have any sync groups'. Below that is the 'Sync Agent' section, which also displays a message: 'You do not have any agents'. At the top right, there are 'New Sync Group' and 'Refresh' buttons.

Specify the following options of the new database

Option	Value
<b>Sync Group Name</b>	<input type="text" value="MySyncGroup"/>
<b>Sync Metadata Database</b>	New Database
<b>Create New Database</b>	Name: <input type="text" value="DataSync"/> Use existing Resource Group Target server: (default your logical server) Pricing Tier: Standard S0 Collation: SQL_Latin1_General_CI_AS
<b>Automatic Sync</b>	On
<b>Sync Frequency</b>	5 minutes
<b>Conflict Resolution</b>	Hub Win

7. Click the **OK** button

Wait until the sync group is created

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 2: Add Sync Members

---

This exercise shows how to add Sync Members

---

### Tasks

- 1. Go to the resource created and Add an Azure Database on the same logical server

Click on **Add an Azure Database**

Specify the following options of the new Sync member

Option	Value
Name	<input type="text" value="SalesdbMemberLocal"/>
<b>Subscription</b>	
<b>Azure SQL Server</b>	
Azure SQL Database	<input type="text" value="SalesdbSync"/>
Sync Directions	Bi-Directional Sync
Username	<input type="text" value="AzureAdmin"/>
Password	<input type="text" value="Passw0rd1"/>

Click **OK** and wait until the member has been added

- 2. Add a second Azure Database on the logical server

Click on **Add an Azure Database**

Specify the following options of the new Sync member

Option	Value
Name	<input type="text" value="SalesdbMemberRemote"/>
<b>Subscription</b>	
<b>Azure SQL Server</b>	

Option	Value
Azure SQL Database	<input type="text"/> SalesdbSync
Sync Directions	Bi-Directional Sync
Username	<input type="text"/> AzureAdmin
Password	<input type="text"/> Passw0rd1

Click **OK** and wait until the member has been added

The result should look like

#### Member Database

[Add an Azure Database](#)



SaledbMemberLocal



SalesdbMemberRemote

[Add an On-Premises Database](#)

No On-Premises database available

- 3. Click **OK** to proceed to Exercise 3 to add the tables

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 3: Add Tables

Before the synchronization can start, you need to specify the tables that you want to synchronize.

### Tasks

- 1. Open the drop down box "Select a database" and select a database - **Hub database**
- 2. Wait until all the tables appear.

**i** If the tables do not appear, click the **Refresh Schema** button

- 3. Select all the tables to sync
- 4. Click **Save**

**Tables**

The screenshot shows the 'Tables' configuration window. At the top, there's a dropdown menu set to 'Hub Database' and a 'Refresh Schema' button. Below this, two sections are visible: 'Select tables to sync' and 'Select fields to sync'. In the 'Select tables to sync' section, three tables are listed: 'dbo.Customers' (4 columns), 'dbo.Employees' (4 columns), and 'dbo.Products' (3 columns). All three tables have their checkboxes checked. In the 'Select fields to sync' section, all fields for each table are listed: CustomerID, FirstName, MiddleInitial, LastName for 'dbo.Customers'; EmployeeID, FirstName, MiddleInitial, LastName for 'dbo.Employees'; and ProductID, Name, Price for 'dbo.Products'. The 'Primary Key' status is indicated for some fields.

NAME	COLUMNS
dbo.Customers	4
dbo.Employees	4
dbo.Products	3

NAME	DATA TYPE	DESCRIPTIONS
CustomerID	int(4)	Primary Key
FirstName	nvarchar(40)	
MiddleInitial	nvarchar(40)	
LastName	nvarchar(40)	
EmployeeID	int(4)	Primary Key
FirstName	nvarchar(40)	
MiddleInitial	nvarchar(40)	
LastName	nvarchar(40)	
ProductID	int(4)	Primary Key
Name	nvarchar(50)	
Price	money(8)	

- 5. Click on **MySyncGroup** and then the **Refresh Logs** button and verify if the initial sync was OK

MySyncGroup  
Database Sync Group

Sync Stop Properties Delete Filter Logs Refresh Logs

Databases Tables

3 SQL 3 ≡

**Logs**

TYPE	DATE/TIME	MEMBER DATABASE	DETAILS
Success	22/03/19 11:34:56	salesdbSync/pvhnordic....	Sync completed successfully in 6.81 seconds. Upload: 0 changes...
Success	22/03/19 11:34:56	salesdbSync/pvhv12.da...	Sync completed successfully in 6.56 seconds. Upload: 0 changes...
Success	22/03/19 11:34:48	salesdbSync/pvhnordic....	Database provisioned successfully in 8.64 seconds.
Success	22/03/19 11:34:47	salesdbSync/pvhv12.da...	Database provisioned successfully in 7.94 seconds.
Success	22/03/19 11:34:47	salesdb/pvhv12.databa...	Database provisioned successfully in 7.36 seconds.
Success	22/03/19 11:34:46	salesdb/pvhv12.databa...	Database provisioned successfully in 7.06 seconds.
Success	22/03/19 11:31:13	salesdb/pvhv12.databa...	Schema information obtained successfully.

- 6. Go to the SQL Server Management Studio (**SSMS**) and verify if the tables are synced to the **SalesdbSync** databases

Congratulations!

You have successfully completed this exercise. Click **Next** to advance to the next exercise.

## Exercise 4: Make modifications and manually sync

---

This exercise shows how to make modifications and manually sync

---

### Tasks

- 1. Insert record into the **Customers** table of the Hub Database

Open the SQL Server Management Studio (**SSMS**) and connect to your Hub Database

**i** This will be the **Salesdb** on the current **primary** logical SQL server

Insert a record in the **Customers** table

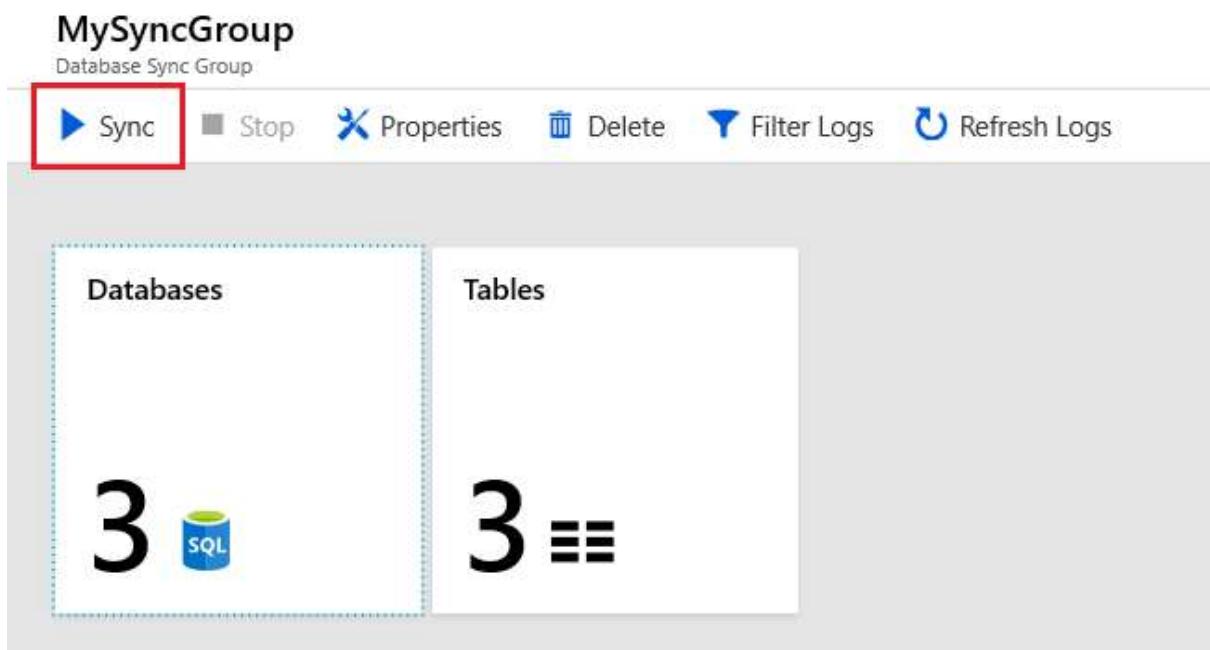
SQL

▶ `INSERT INTO Customers([FirstName], [MiddleInitial], [LastName])  
VALUES ('Brandon','','Flowers')`

**i** The script can also be found in the **C:\LabFiles\M06L06Lab01** folder of the virtual machine

- 2. Manually start a synchronization

Connect to the Azure Portal, go to your sync group and press the **Sync** Button



- 3. Verify sync

Connect to the **SalesdbSync** database on the same logical server and execute the following statement

SQL

```
▶ SELECT [CustomerID]
    ,[FirstName]
    ,[MiddleInitial]
    ,[LastName]
  FROM [dbo].[Customers]
 WHERE LastName='Flowers'
```

The result should be

	CustomerID	FirstName	MiddleInitial	LastName
1	1001	Brandon		Flowers

Connect to the **SalesdbSync** database on the DR logical server and execute the same statement.  
The result should be the same.

- 4. Update record into the **Customers** table of the **Member** Database

Connect to the **SalesdbSync** database on the DR logical server and execute the following statement:

SQL

```
▶ UPDATE Customers
  SET FirstName='Lenny', LastName='Kravitz'
 WHERE LastName='Flowers'
```

- 5. Manually start a synchronization

Connect to the Azure Portal, go to your sync group and press the **Sync** Button

- 6. Verify sync

Connect to the **SalesdbSync** and **salesdb** database on your logical server and execute the following statement

SQL

```
▶ SELECT [CustomerID]
    ,[FirstName]
    ,[MiddleInitial]
    ,[LastName]
  FROM [dbo].[Customers]
 WHERE LastName='Kravitz'
```

- On the **salesdb** you should see 1 record returned, on the **salesdbsync** no records will be returned. The reason is that the **member** database is first synced with the **hub** database. Secondly, the **hub** database will sync to the other member databases. If you start the Sync manually again, you should see the same record available in the **salesdbsync** DB as well.
- 7. Close all open windows without saving.

Congratulations!

You have successfully completed this lab. To mark the lab as complete, click **End**.

