# Full Stack Application Development with MS Azure Cloud

## Module 6 – Application Deployment and Management with Azure

## Lab Practical Manual

### Topic: Azure SQL DB – Solved Question

**Lab 1: Create a database using SQL on AZURE.**

### Create an Azure SQL Database single database

In this quickstart, you create a single database in Azure SQL Database using either the Azure portal, a PowerShell script, or an Azure CLI script. You then query the database using **Query editor** in the Azure portal.

#### Prerequisite

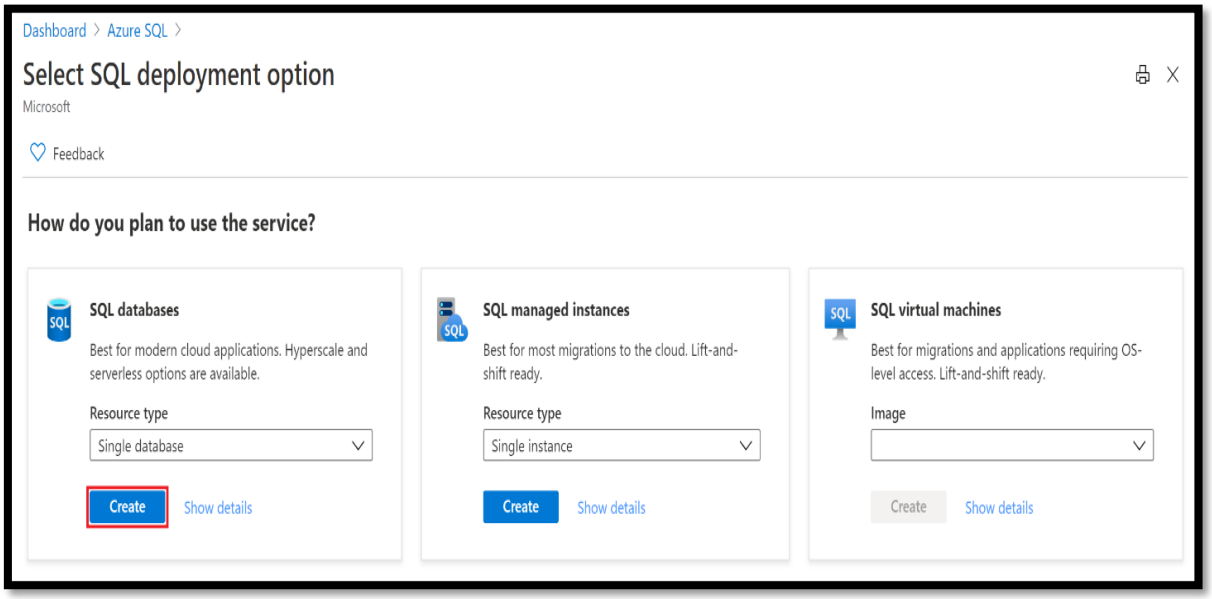
- An active Azure subscription. If you don't have one, create a free account.

#### Create a single database

This quickstart creates a single database in the serverless compute tier.

To create a single database in the Azure portal this quickstart starts at the Azure SQL page.

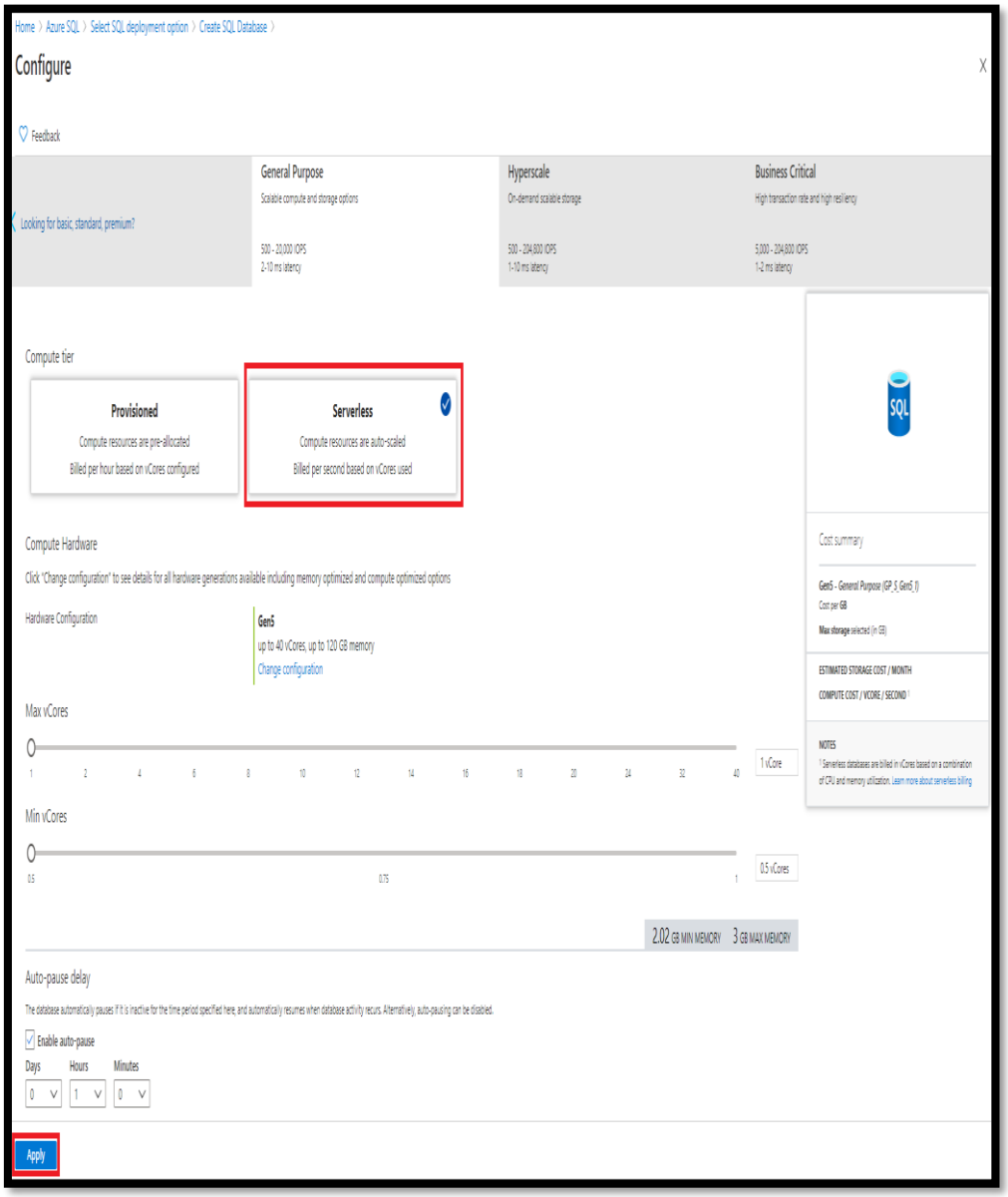
1. Browse to the Select SQL Deployment option page.
2. Under **SQL databases**, leave **Resource type** set to **Single database**, and select **Create**.



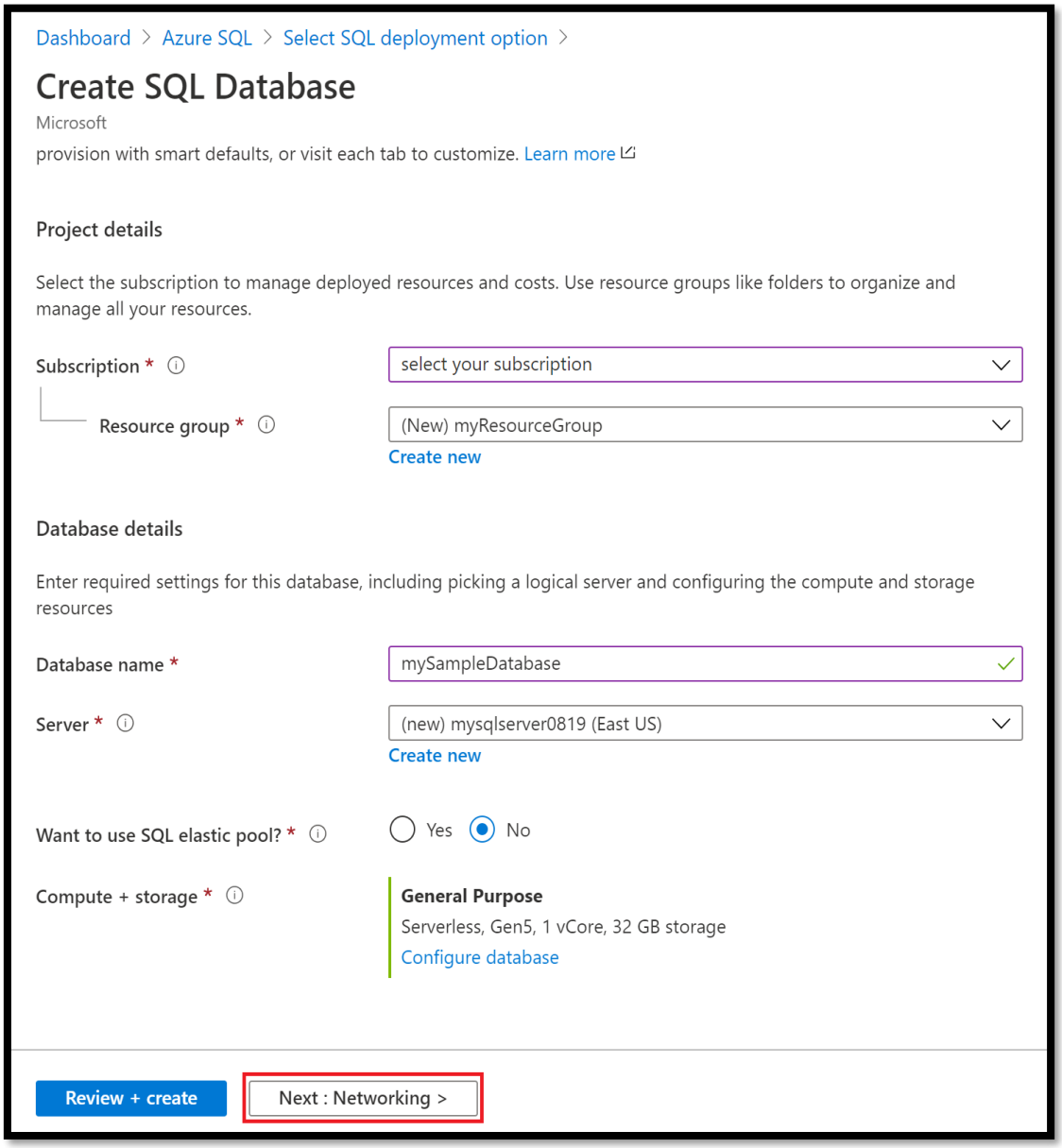

3. On the **Basics** tab of the **Create SQL Database** form, under **Project details**, select the desired Azure **Subscription**.
4. For **Resource group**, select **Create new**, enter *myResourceGroup*, and select **OK**.
5. For **Database name** enter *mySampleDatabase*.
6. For **Server**, select **Create new**, and fill out the **New server** form with the following values:

- **Server name**: Enter *mysqlserver*, and add some characters for uniqueness. We can't provide an exact server name to use because server names must be globally unique for all servers in Azure, not just unique within a subscription. So enter something like mysqlserver12345, and the portal lets you know if it is available or not.
- **Server admin login**: Enter *azureuser*.
- **Password**: Enter a password that meets requirements, and enter it again in the **Confirm password** field.
- **Location**: Select a location from the dropdown list.

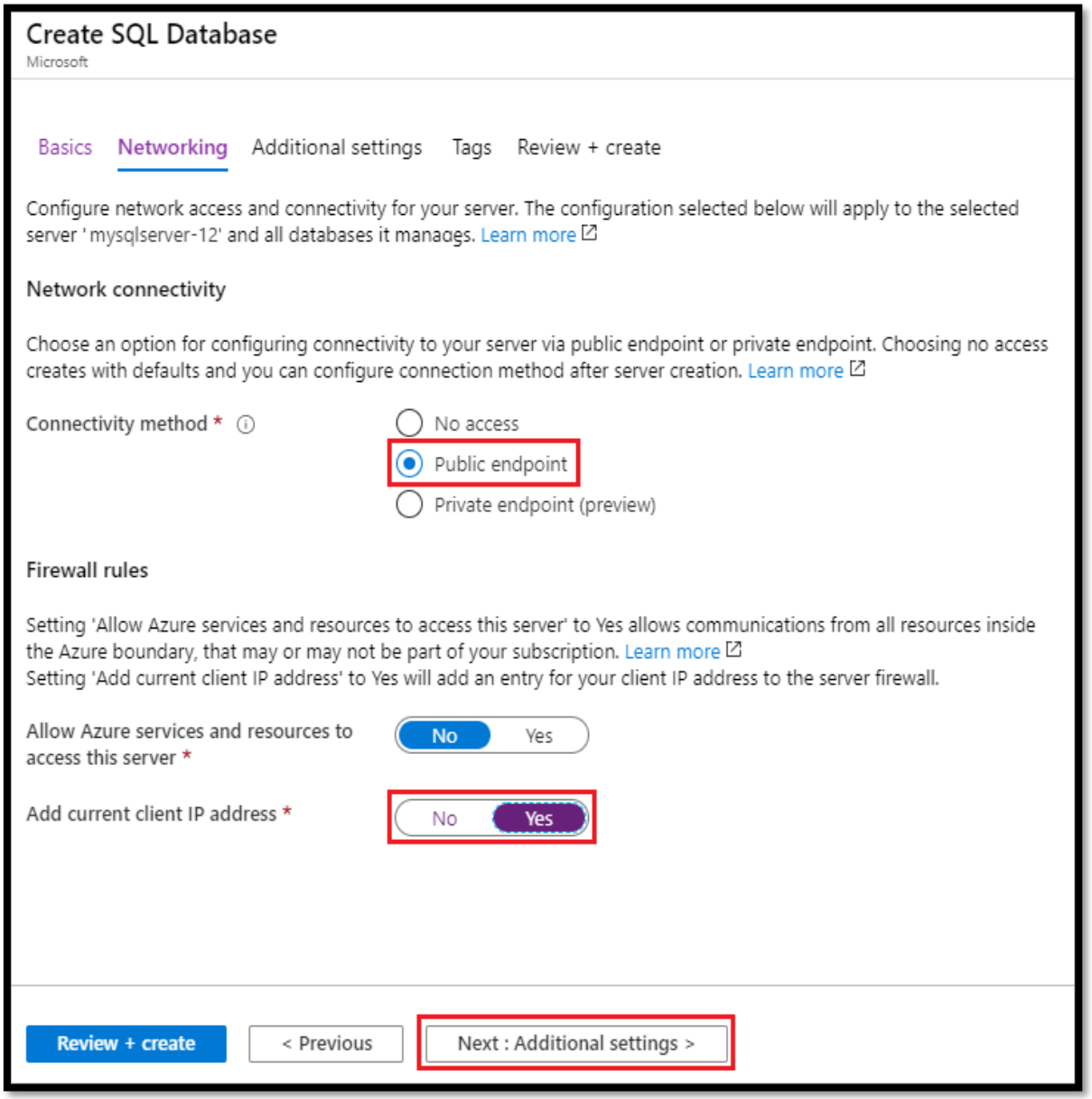
  Select **OK**.



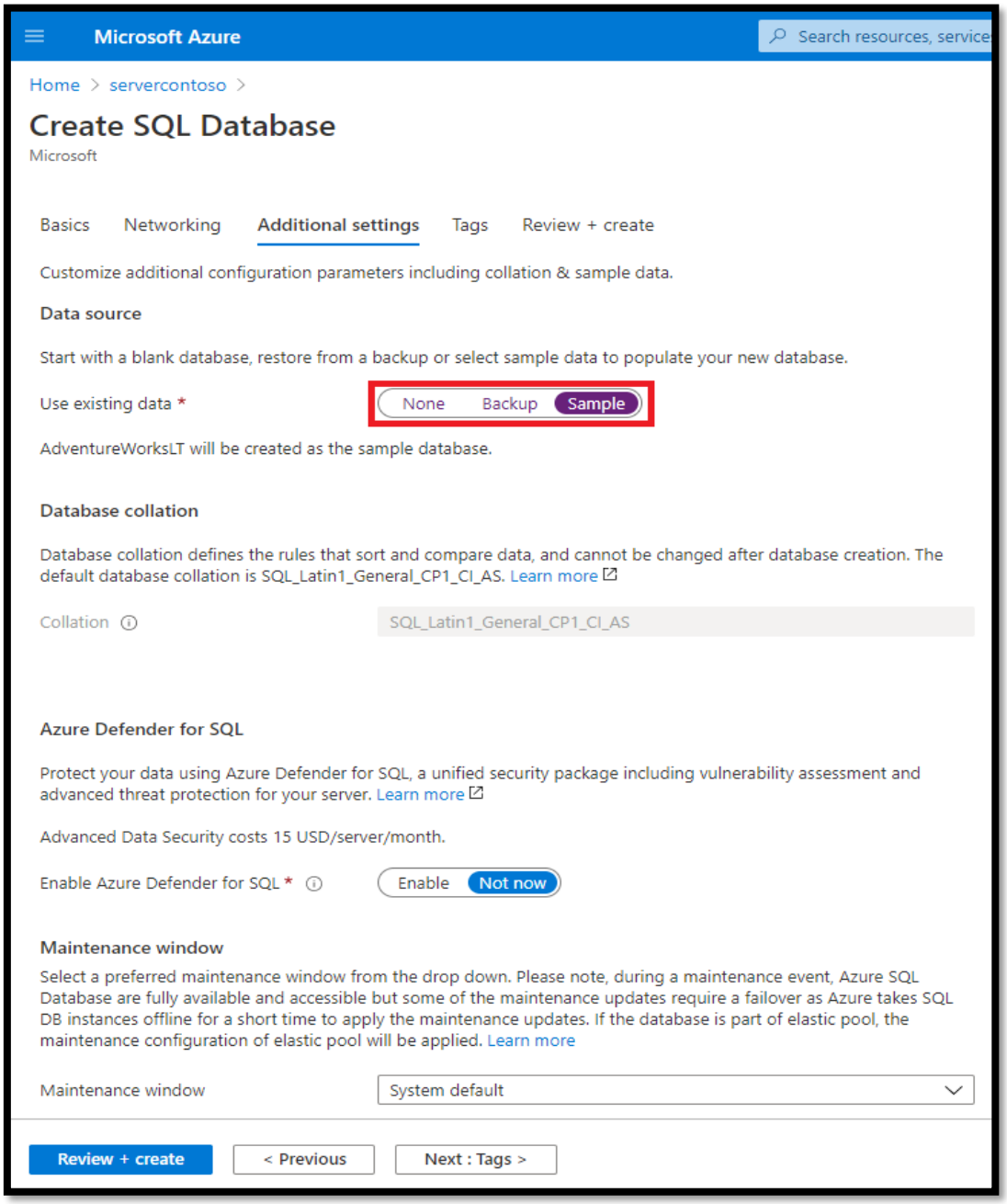

7. Leave **Want to use SQL elastic pool** set to **No**.
8. Under **Compute + storage**, select **Configure database**.
9. This quickstart uses a serverless database, so select **Serverless**, and then select **Apply**.
10. Select **Next: Networking** at the bottom of the page.




11. On the **Networking** tab, for **Connectivity method**, select **Public endpoint**.
12. For **Firewall rules**, set **Add current client IP address** to **Yes**. Leave **Allow Azure services and resources to access this server** set to **No**.
13. Select **Next: Additional settings** at the bottom of the page.




14. On the **Additional settings** tab, in the **Data source** section, for **Use existing data**, select **Sample**. This creates an AdventureWorksLT sample database so there's some tables and data to query and experiment with, as opposed to an empty blank database.
15. Optionally, enable Azure Defender for SQL.
16. Optionally, set the maintenance window so planned maintenance is performed at the best time for your database.
17. Select **Review + create** at the bottom of the page:

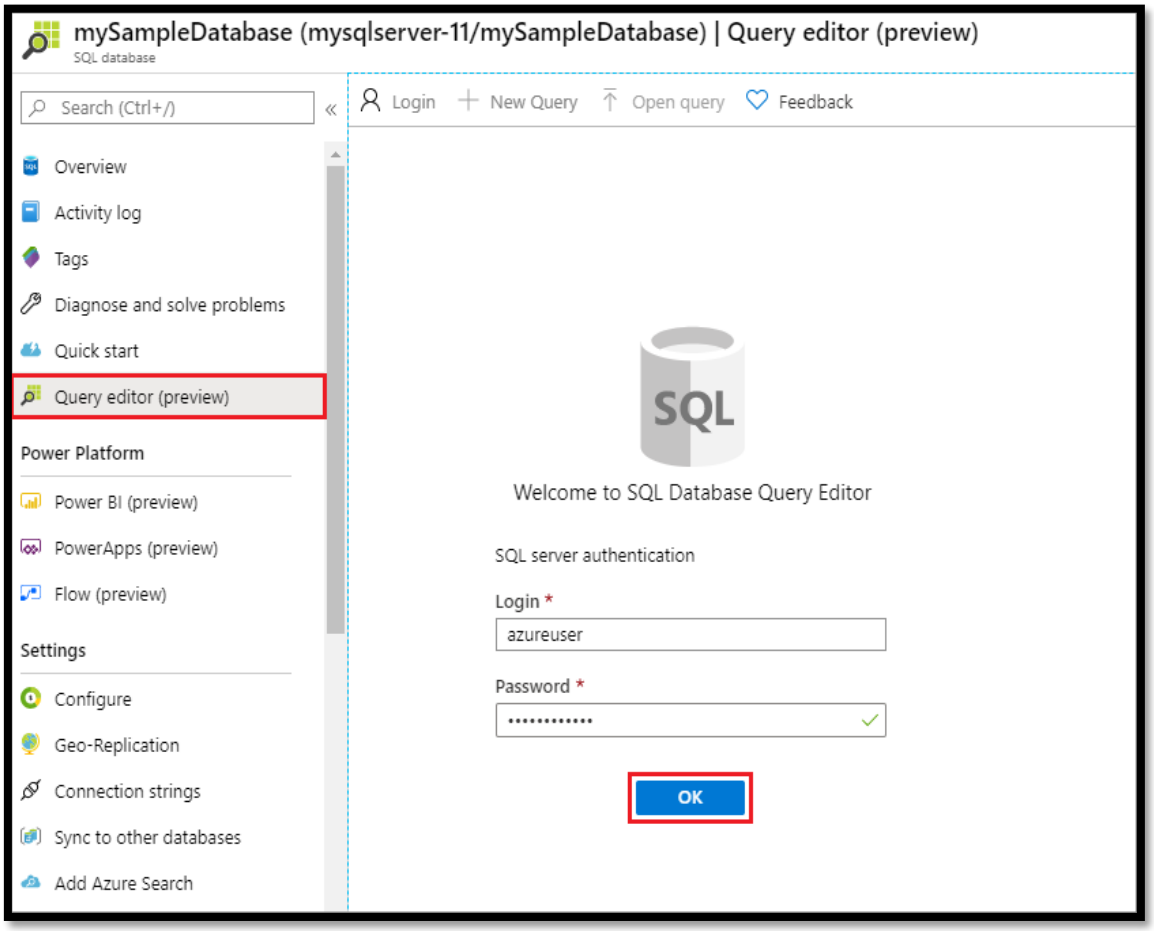



18. On the **Review + create** page, after reviewing, select **Create**.

## Query the database

Once your database is created, you can use the **Query editor (preview)** in the Azure portal to connect to the database and query data.

1. In the portal, search for and select **SQL databases**, and then select your database from the list.
2. On the page for your database, select **Query editor (preview)** in the left menu.
3. Enter your server admin login information, and select **OK**.

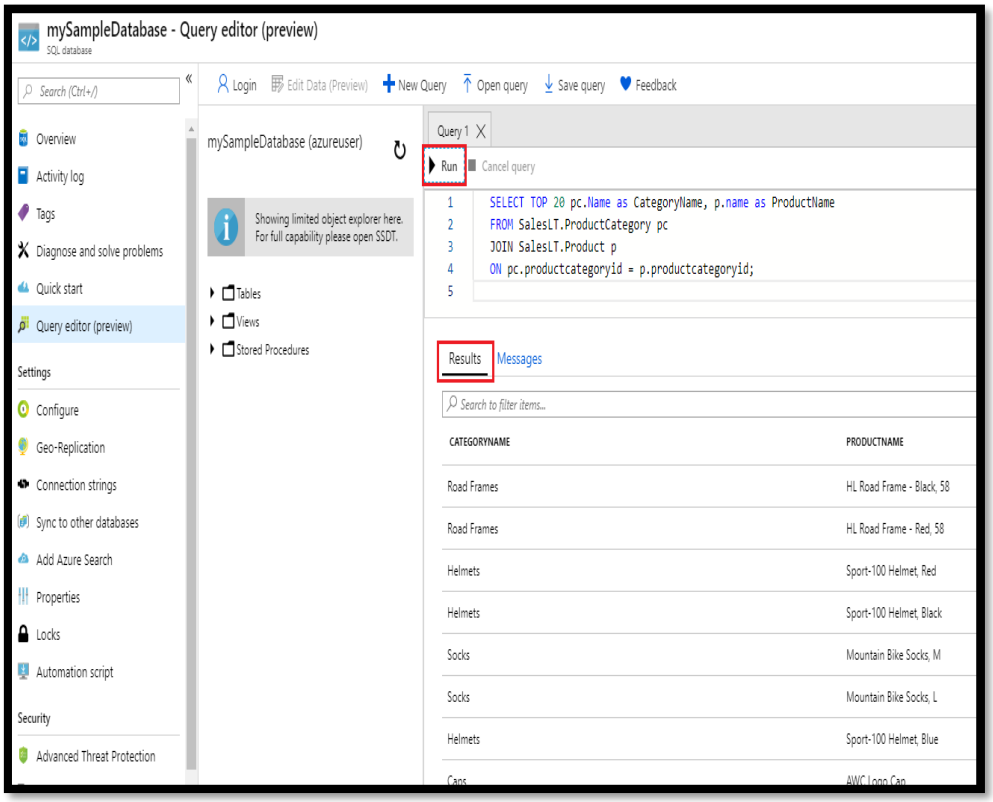



4. Enter the following query in the **Query editor** pane.

   SQLCopy

   ```
   SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
   FROM SalesLT.ProductCategory pc
   JOIN SalesLT.Product p
   ON pc.productcategoryid = p.productcategoryid;
   ```

5. Select **Run**, and then review the query results in the **Results** pane.

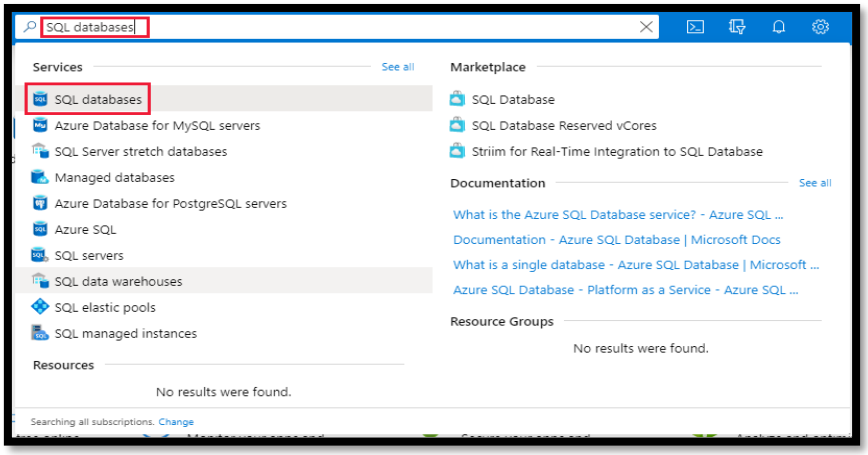



6. Close the **Query editor** page, and select **OK** when prompted to discard your unsaved edits.

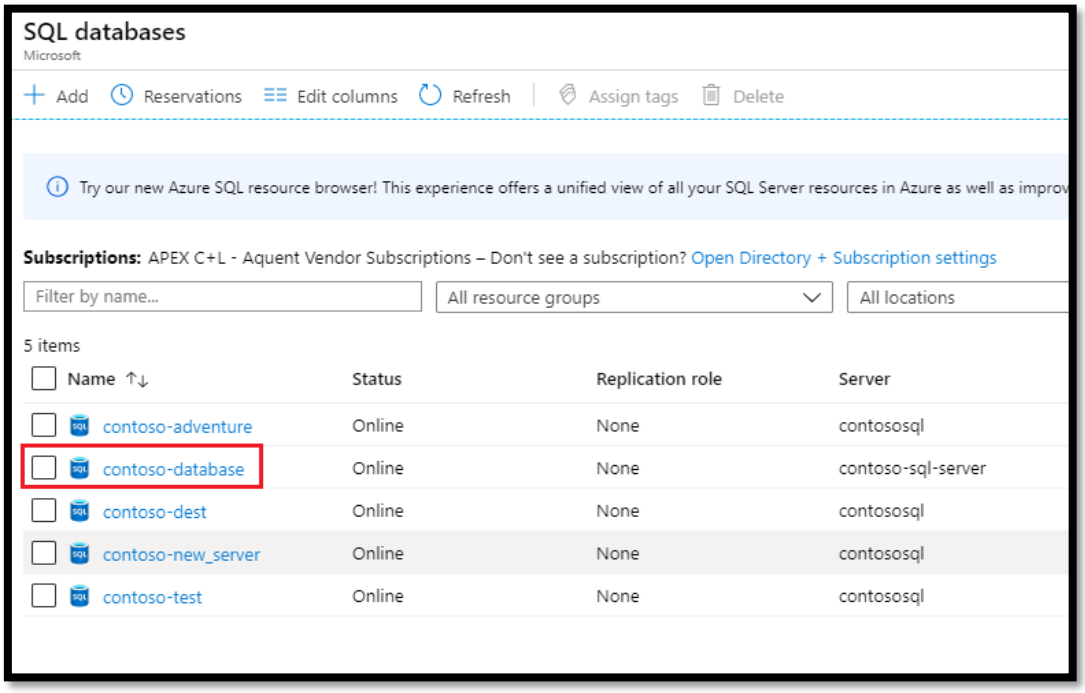
### 3.6.2 Set up SQL Data Sync between databases in Azure SQL Database and SQL Server

#### Create sync group

1. Go to the Azure portal to find your database in SQL Database. Search for and select **SQL databases**.

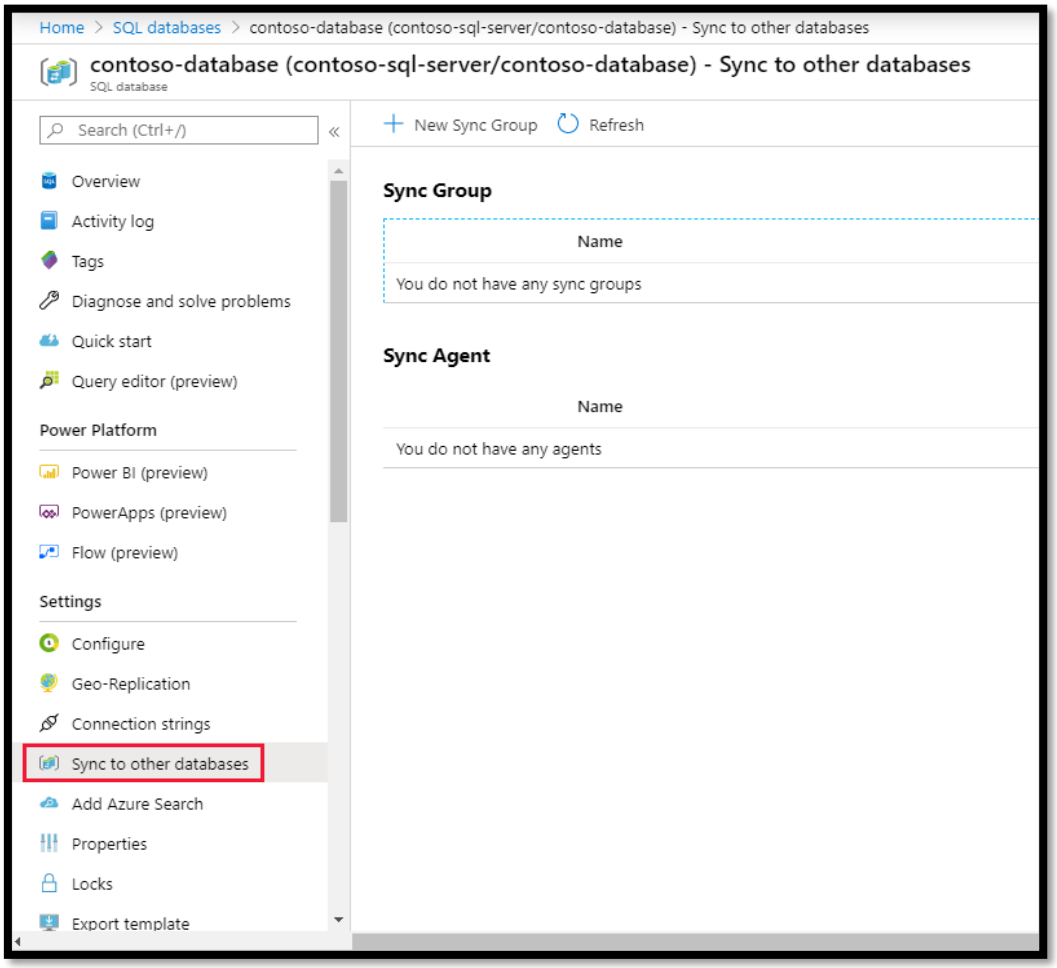


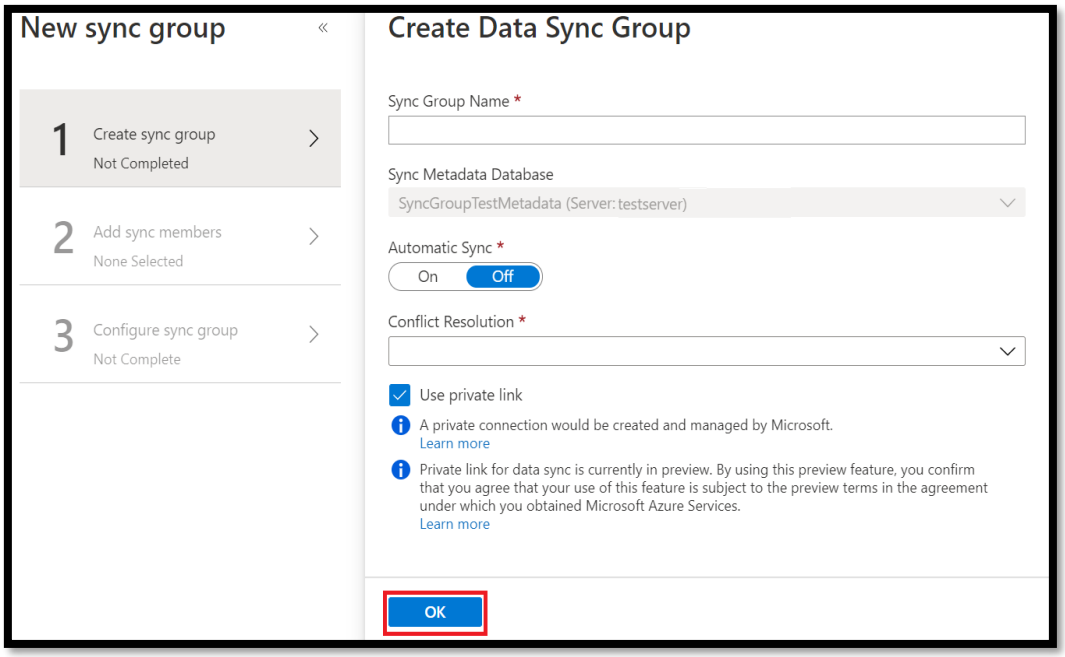

2. Select the database you want to use as the hub database for Data Sync.




**Note:** The hub database is a sync topology's central endpoint, in which a sync group has multiple database endpoints. All other member databases with endpoints in the sync group, sync with the hub database.

3. On the **SQL database** menu for the selected database, select **Sync to other databases**.




On the **Create Data Sync Group** page, change the following settings:

4. On the **Sync to other databases** page, select **New Sync Group**. The **New sync group** page opens with **Create sync group (step 1)**.

**ABLE 25**

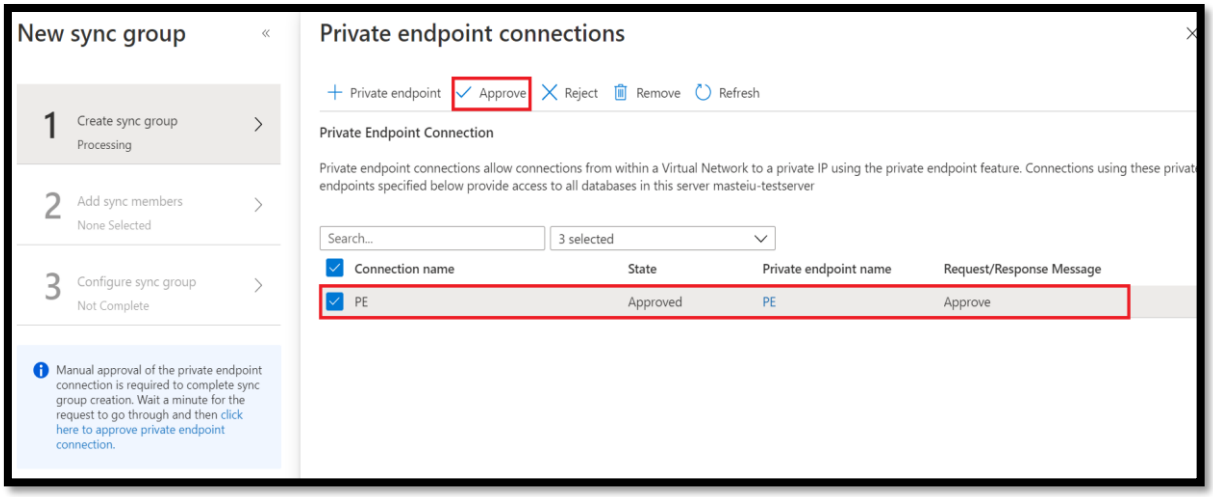
| **etting** | **escription** |
|---|---|
| **ync Group ame** | nter a name for the new sync group. This name is distinct from the ame of the database itself. |
| **ync Metadata atabase** | hoose to create a database (recommended) or to use an existing atabase.<br><br>you choose **New database**, select **Create new database.** Then on e **SQL Database** page, name and configure the new database and elect **OK**.<br><br>you choose **Use existing database**, select the database from the t. |
| **utomatic Sync** | elect **On** or **Off**.<br><br>you choose **On**, enter a number and elect **Seconds**, **Minutes**, **Hours**, or **Days** in the **Sync requency** section.<br>he first sync begins after the selected interval period elapses from the me the configuration is saved. |
| **onflict esolution** | elect **Hub win** or **Member win**.<br><br>**ub win** means when conflicts occur, data in the hub database verwrites conflicting data in the member database.<br><br>**ember win** means when conflicts occur, data in the member atabase overwrites conflicting data in the hub database. |
| **se private link** | hoose a service managed private endpoint to establish a secure onnection between the sync service and the hub database. |

**Note:** Microsoft recommends to create a new, empty database for use as the **Sync Metadata Database**. Data Sync creates tables in this database and runs a frequent workload. This database is shared as the **Sync Metadata Database** for all sync groups in a selected region and subscription. You can't change the database or its name without removing all sync groups and sync agents in the region. Additionally,

an Elastic jobs database cannot be used as the SQL Data Sync Metadata database and vice versa.

Select **OK** and wait for the sync group to be created and deployed.

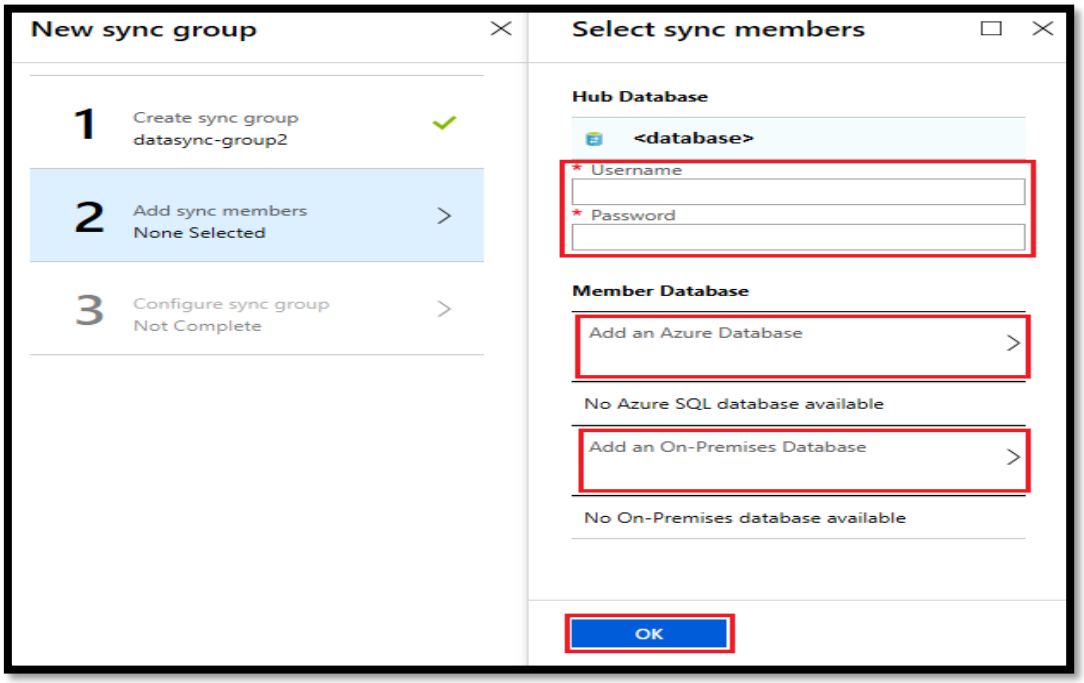
5. On the **New Sync Group** page, if you selected **Use private link**, you will need to approve the private endpoint connection. The link in the info message will take you to the private endpoint connections experience where you can approve the connection.

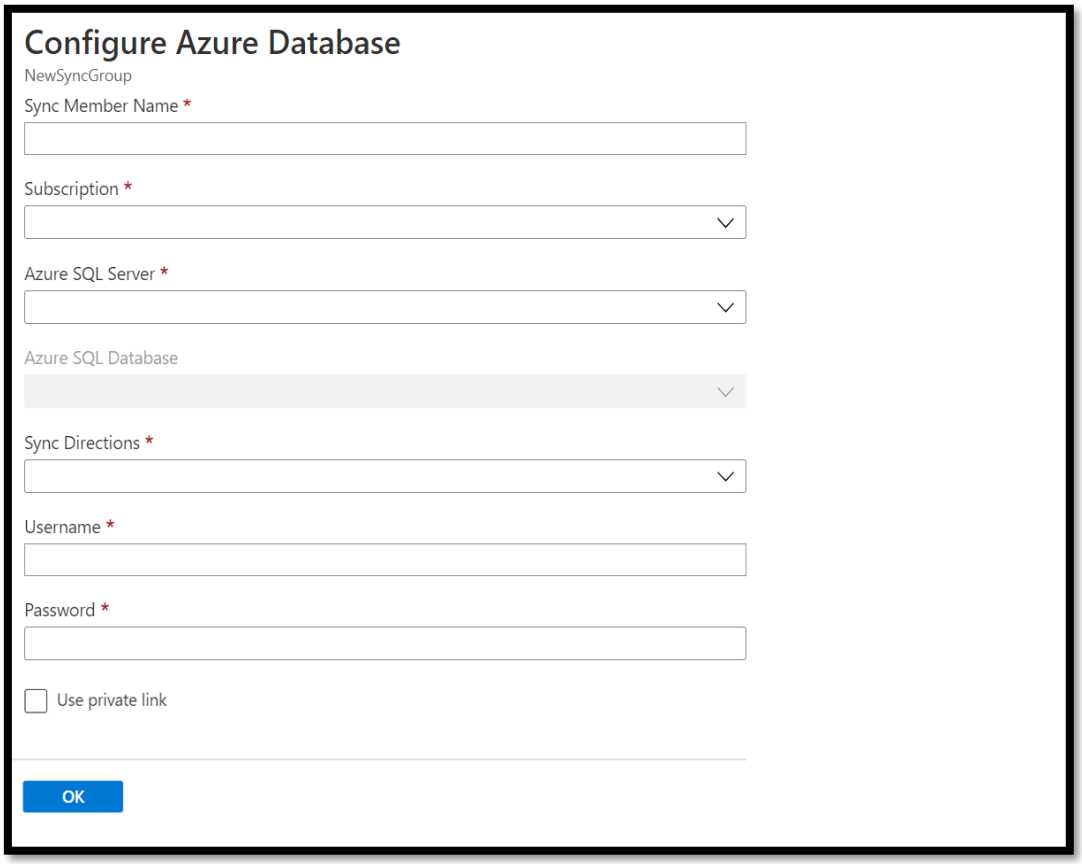



## Add sync members

After the new sync group is created and deployed, **Add sync members (step 2)** is highlighted on the **New sync group** page.

In the **Hub Database** section, enter existing credentials for the server on which the hub database is located. Don't enter *new* credentials in this section.


New sync group
1
Create sync group
datasync-group2
2
Add sync members
None Selected
3
Configure sync group
Not Complete
Select sync members
Hub Database
<database>
* Username
* Password
Member Database
Add an Azure Database
No Azure SQL database available
Add an On-Premises Database
No On-Premises database available
OK


Configure Azure Database
NewSyncGroup
Sync Member Name *
Subscription *
Azure SQL Server *
Azure SQL Database
Sync Directions *
Username *
Password *
Use private link
OK

### To add a database in Azure SQL Database

In the **Member Database** section, optionally add a database in Azure SQL Database to the sync group by selecting **Add an Azure SQL Database**. The **Configure Azure SQL Database** page opens.

On the **Configure Azure SQL Database** page, change the following settings:

**ABLE 26:TO ADD A DATABASE IN AZURE SQL DATABASE**

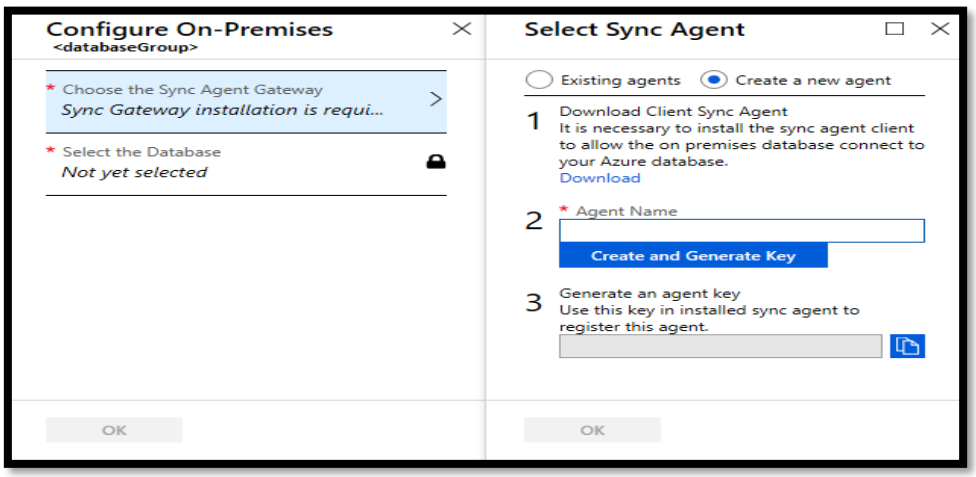
| **etting** | **escription** |
|---|---|
| **ync Member Name** | rovide a name for the new sync member. This name is stinct from the database name itself. |
| **ubscription** | elect the associated Azure subscription for billing urposes. |
| **zure SQL Server** | elect the existing server. |
| **zure SQL Database** | elect the existing database in SQL Database. |
| **ync Directions** | elect **Bi-directional Sync**, **To the Hub**, or **From the Hub**. |
| **sername** and **Password** | nter the existing credentials for the server on which the ember database is located. Don't enter *new* credentials in is section. |
| **se private link** | hoose a service managed private endpoint to establish a ecure connection between the sync service and the ember database. |

Select **OK** and wait for the new sync member to be created and deployed.

To add a SQL Server database

In the **Member Database** section, optionally add a SQL Server database to the sync group by selecting **Add an On-Premises Database**. The **Configure On-Premises** page opens where you can do the following things:

1. Select **Choose the Sync Agent Gateway**. The **Select Sync Agent** page opens.




2. On the **Choose the Sync Agent** page, choose whether to use an existing agent or create an agent.

   If you choose **Existing agents**, select the existing agent from the list.

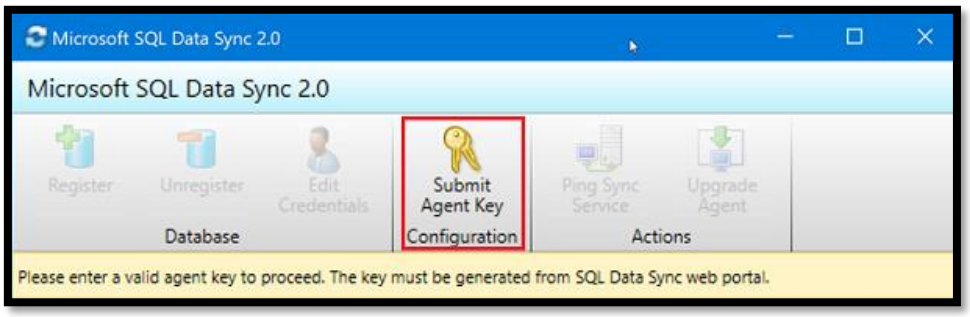
   If you choose **Create a new agent**, do the following things:

1. Download the data sync agent from the link provided and install it on the computer where the SQL Server is located. You can also download the agent directly from Azure SQL Data Sync Agent.

   **Important**

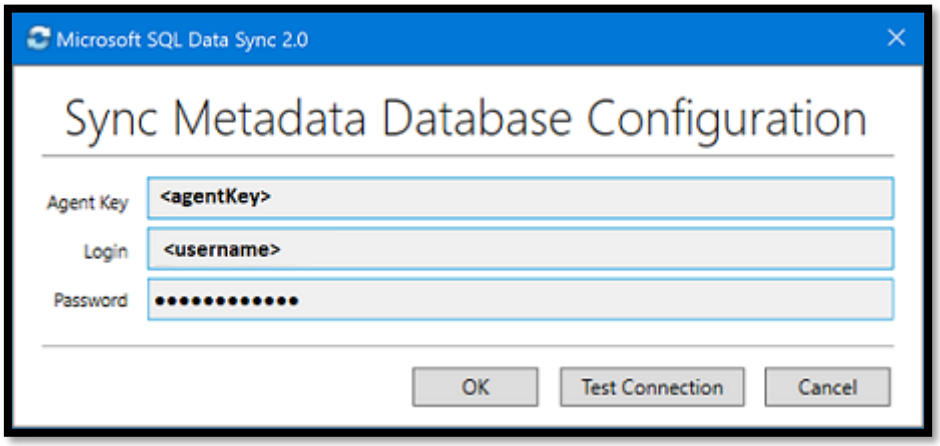
   You have to open outbound TCP port 1433 in the firewall to let the client agent communicate with the server.

2. Enter a name for the agent.
3. Select **Create and Generate Key** and copy the agent key to the clipboard.
4. Select **OK** to close the **Select Sync Agent** page.
3. On the SQL Server computer, locate and run the Client Sync Agent app.



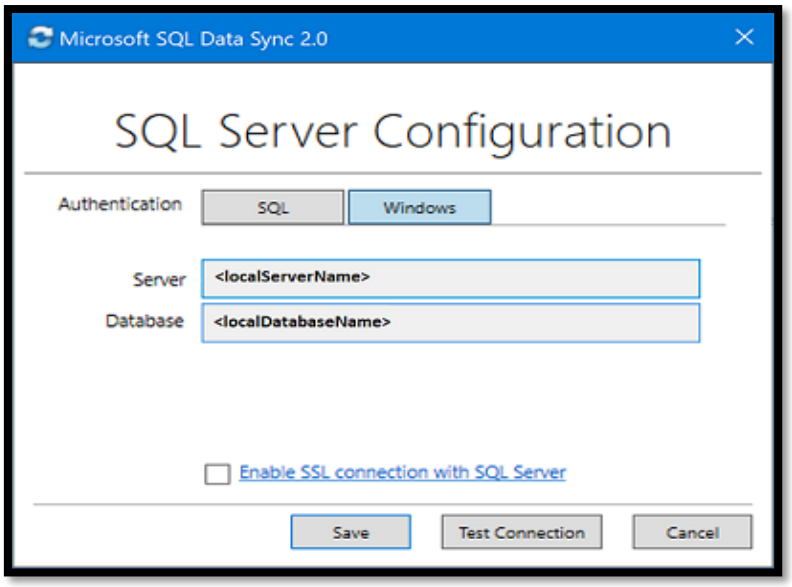

1. In the sync agent app, select **Submit Agent Key**. The **Sync Metadata Database Configuration** dialog box opens.

2. In the **Sync Metadata Database Configuration** dialog box, paste in the agent key copied from the Azure portal. Also provide the existing credentials for the server on which the metadata database is located. (If you created a metadata database, this database is on the same server as the hub database.) Select **OK** and wait for the configuration to finish.



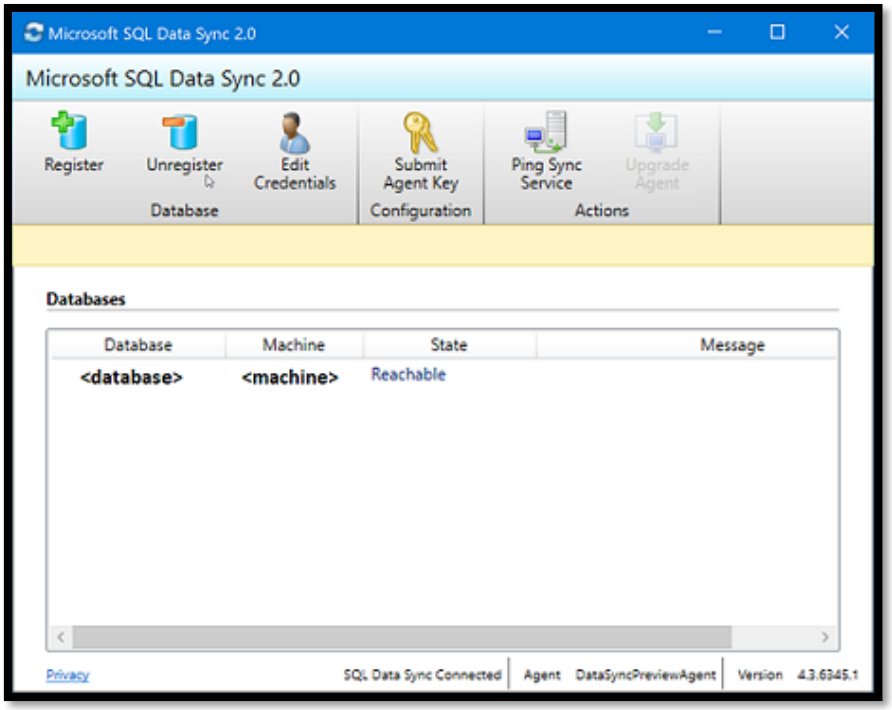

**Note-** If you get a firewall error, create a firewall rule on Azure to allow incoming traffic from the SQL Server computer. You can create the rule manually in the portal or in SQL Server Management Studio (SSMS). In SSMS, connect to the hub database on Azure by entering its name as

<hub_database_name>.database.windows.net.

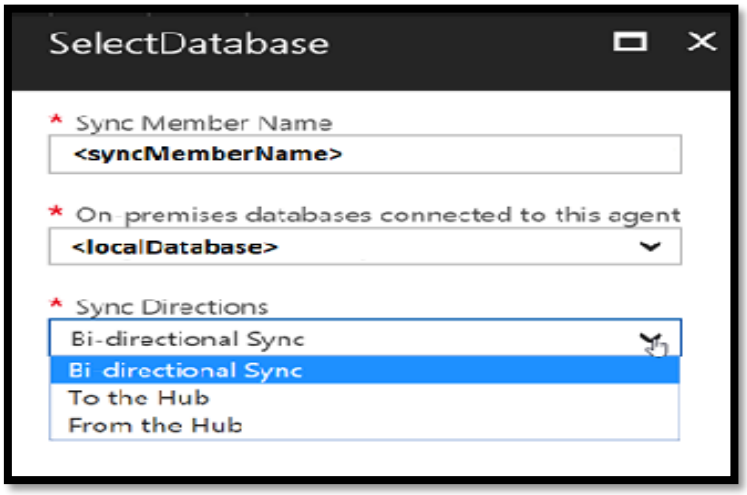



3. Select **Register** to register a SQL Server database with the agent. The **SQL Server Configuration** dialog box opens.
4. In the **SQL Server Configuration** dialog box, choose to connect using SQL Server authentication or Windows authentication. If you choose SQL Server authentication,

enter the existing credentials. Provide the SQL Server name and the name of the database that you want to sync and select **Test connection** to test your settings. Then select **Save** and the registered database appears in the list.



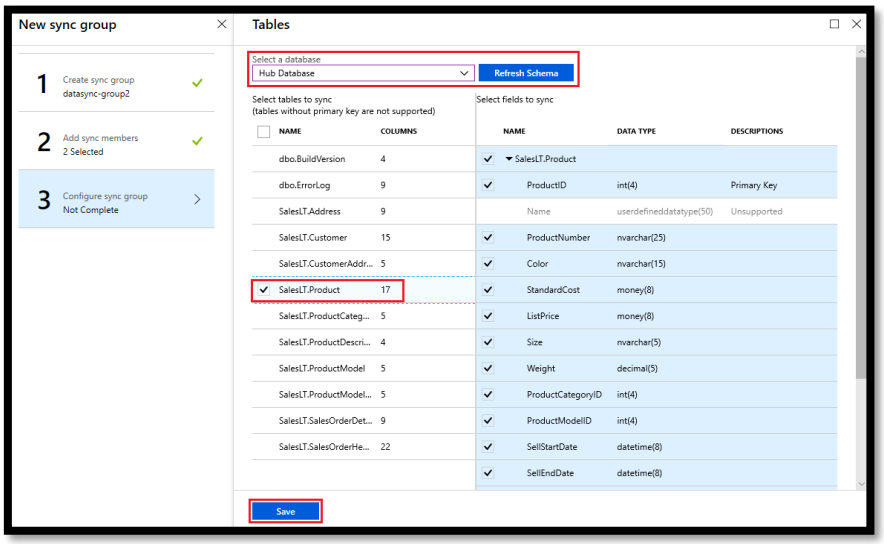

1. Close the Client Sync Agent app.
2. In the portal, on the **Configure On-Premises** page, select **Select the Database**.
3. On the **Select Database** page, in the **Sync Member Name** field, provide a name for the new sync member. This name is distinct from the name of the database itself. Select the database from the list. In the **Sync Directions** field, select **Bi-directional Sync**, **To the Hub**, or **From the Hub**.
4. Select **OK** to close the **Select Database** page. Then select **OK** to close the **Configure On-Premises** page and wait for the new sync member to be created and deployed. Finally, select **OK** to close the **Select sync members** page.




Note: To connect to SQL Data Sync and the local agent, add your user name to the role *DataSync_Executor*. Data Sync creates this role on the SQL Server instance.

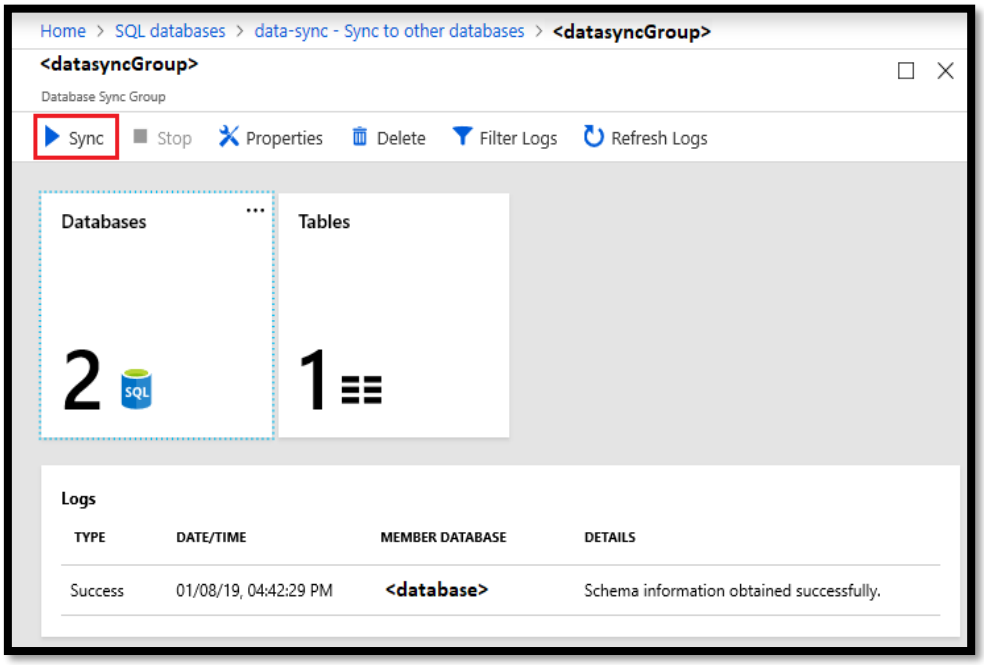
## Configure sync group

After the new sync group members are created and deployed, **Configure sync group (step 3)** is highlighted in the **New sync group** page.




1. On the **Tables** page, select a database from the list of sync group members and select **Refresh schema**.
2. From the list, select the tables you want to sync. By default, all columns are selected, so disable the checkbox for the columns you don't want to sync. Be sure to leave the primary key column selected.
3. Select **Save**.
4. By default, databases are not synced until scheduled or manually run. To run a manual sync, navigate to your database in SQL Database in the Azure portal,

select **Sync to other databases**, and select the sync group. The **Data Sync** page opens. Select **Sync**.




Activity: This practical activity gives a quick start to learner about relational database through SQL commands. User will create a single database and perform SQL operations using SQL DDL, DML commands to create database tables and insert data. Also, data can be updated and truncated later. This gives idea of how cloud relational databases can be worked with.