Create a Virtual Network

1. In the Azure portal home page, select the left hand menu.
2. In the left navigation pane, select **Virtual Networks**
3. Select **+ Create** to open the **Create Virtual Network** page. On the **Basics** tab, complete the following information:
    - **Subscription:** <Your subscription>
    - **Resource group:** starting with *DP300* or the resource group you previously selected
    - **Name:** lab02-vnet
    - **Region:** Select the same region where your resource group was created
4. Select **Review + Create**, review the settings for the new virtual network, and then select **Create**.

Provision an Azure SQL Database in the Azure portal

1. From the Azure Portal, search for *SQL databases* in the search box at the top, then select **SQL databases** from the list of options.
2. On the **SQL databases** blade, select **+ Create**.
3. On the **Create SQL Database** page, select the following options on the **Basics** tab and then select **Next: Networking**.
    - **Subscription:** <Your subscription>
    - **Resource group:** starting with *DP300* or the resource group you previously selected
    - **Database Name:** AdventureWorksLT
    - **Server:** select on **Create new** link. The **Create SQL Database Server** page will open. Provide the server details as follow:
        - **Server name:** dp300-lab-<your initials (lower case)> and if needed a random 5 digit number (server name must be globally unique)
        - **Location:** <your local region, same as the selected region for your resource group, otherwise it may fail>
        - **Authentication method:** Use SQL authentication

        - **Server admin login:** dp300admin
        - **Password:** select a complex password and take note of it
        - **Confirm password:** select the same previously selected password
    - Select **OK** to to return to the **Create SQL Database** page.
    - **Want to use Elastic Pool?** set to **No**.
    - **Workload environment:** Development
    - On the **Compute + Storage** option, select on **Configure database** link. On the **Configure** page, for **Service tier** dropdown, select **Basic**, and then **Apply**.
4. For the **Backup storage redundancy** option, keep the default value: **Local-redundant backup storage**.
5. Then select **Next: Networking**.
6. On the **Networking** tab, for **Network Connectivity** option, select the **Private endpoint** radio button.
7. Then select the **+ Add private endpoint** link under the **Private endpoints** option.
8. Complete the **Create private endpoint** right pane as follows:
    - **Subscription:** <Your subscription>
    - **Resource group:** starting with *DP300* or the resource group you previously selected
    - **Location:** <your local region, same as the selected region for your resource group, otherwise it may fail>
    - **Name:** DP-300-SQL-Endpoint
    - **Target sub-resource:** SqlServer
    - **Virtual network:** lab02-vnet
    - **Subnet:** lab02-vnet/default (10.x.0.0/24)
    - **Integrate with private DNS zone:** Yes
    - **Private DNS zone:** keep the default value
    - Review settings, and then select **OK**

9. The new endpoint will appear on the **Private endpoints** list.
10. Select **Next: Security**, and then **Next: Additional settings**.
11. On the **Additional settings** page, select **Sample** on the **Use existing data** option. Select **OK** if a pop-up message is displayed for the sample database.
12. Select **Review + Create**.
13. Review the settings before selecting **Create**.
14. Once the deployment is complete, select **Go to resource**.

Enable access to an Azure SQL Database

1. From the **SQL database** page, select the **Overview** section, and then select the link for the server name in the top section.
2. On the SQL servers navigation blade, select **Networking** under the **Security** section.
3. On the **Public access** tab, select **Selected networks**.
4. Select **+ Add your client IPv4 address**. This will add a firewall rule to allow your current IP address to access the SQL server.
5. Check the **Allow Azure services and resources to access this server** property.
6. Select **Save**.

Connect to an Azure SQL Database in SQL Server Management Studio

1. On the Azure portal, select the **SQL databases** in the left navigation pane. And then select the **AdventureWorksLT** database.
2. Copy the **Server name** value from the **Overview** page.
3. Launch SQL Server Management Studio from the lab virtual machine if provided or your local machine if not.
4. In the **Connect to Server** dialog, paste the **Server name** value copied from the Azure portal.
5. In the **Authentication** dropdown, select **SQL Server Authentication**.
6. In the **Login** field, enter **dp300admin**.

7. In the **Password** field, enter the password selected during the SQL server creation.
8. Select **Connect**.
9. SQL Server Management Studio will connect to your Azure SQL Database server. You can expand the server and then the **Databases** node to see the *AdventureWorksLT* database.

Query an Azure SQL Database with SQL Server Management Studio

1. In SQL Server Management Studio, right-click on the *AdventureWorksLT* database and select **New Query**.
2. Paste the following SQL statement into the query window:

sqlTypeCopy

```
SELECT TOP 10 cust.[CustomerID],
  cust.[CompanyName],
  SUM(sohead.[SubTotal]) as OverallOrderSubTotal
FROM [SalesLT].[Customer] cust
  INNER JOIN [SalesLT].[SalesOrderHeader] sohead
    ON sohead.[CustomerID] = cust.[CustomerID]
GROUP BY cust.[CustomerID], cust.[CompanyName]
ORDER BY [OverallOrderSubTotal] DESC
```

3. Select on the **Execute** button in the toolbar to execute the query.
4. In the **Results** pane, review the results of the query.
5. Right-click on the *AdventureWorksLT* database and select **New Query**.
6. Paste the following SQL statement into the query window:

sqlTypeCopy

```
SELECT TOP 10 cat.[Name] AS ProductCategory,
  SUM(detail.[OrderQty]) AS OrderedQuantity
FROM salesLT.[ProductCategory] cat
  INNER JOIN [SalesLT].[Product] prod
```

```
    ON prod.[ProductCategoryID] = cat.[ProductCategoryID]
  INNER JOIN [SalesLT].[SalesOrderDetail] detail
    ON detail.[ProductID] = prod.[ProductID]
GROUP BY cat.[name]
ORDER BY [OrderedQuantity] DESC
```

7. Select on the **Execute** button in the toolbar to execute the query.
8. In the **Results** pane, review the results of the query.
9. Close SQL Server Management Studio. Select **No** when prompted to save changes.