# Lab 1A: Migrating a SQL Server database to SQL Azure

## What you will learn

In this lab, you perform a migration from a SQL 2012 database running on the SQLVM, to SQL Azure PaaS, using SQL Server Management Studio (SSMS):

- Deploy a new SQL Azure server instance;
- Authenticate to SSMS on the SQLVM Virtual Machine;
- Run the database migration wizard from within SSMS;
- Verify the successful migration of the SQL database from the VM to Azure;
- Update the connection strings on the WebVM web application to point to the SQL Azure database instead of the on-premises one on SQLVM;

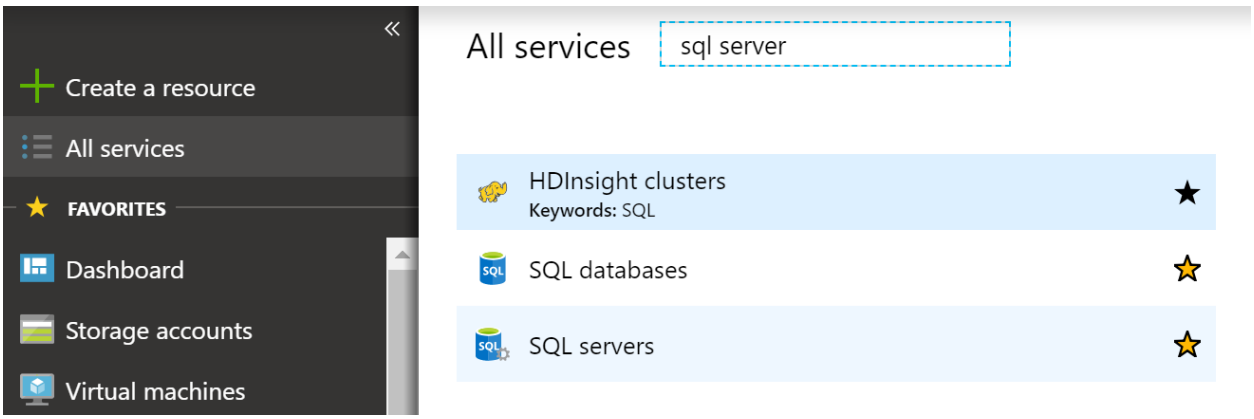
## Time estimate

This lab is estimated to take **60min**, assuming your Azure subscription is already available.

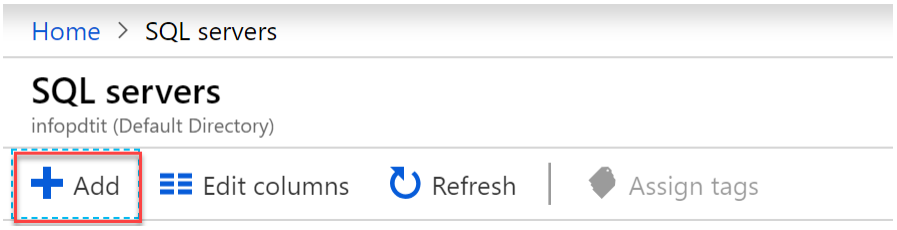
## Task 1: Deploying a new SQL Azure Server instance

In this task, you start from deploying a new SQL Azure Server instance from within the Azure Portal, allowing you to migrate a database to it in the next task.

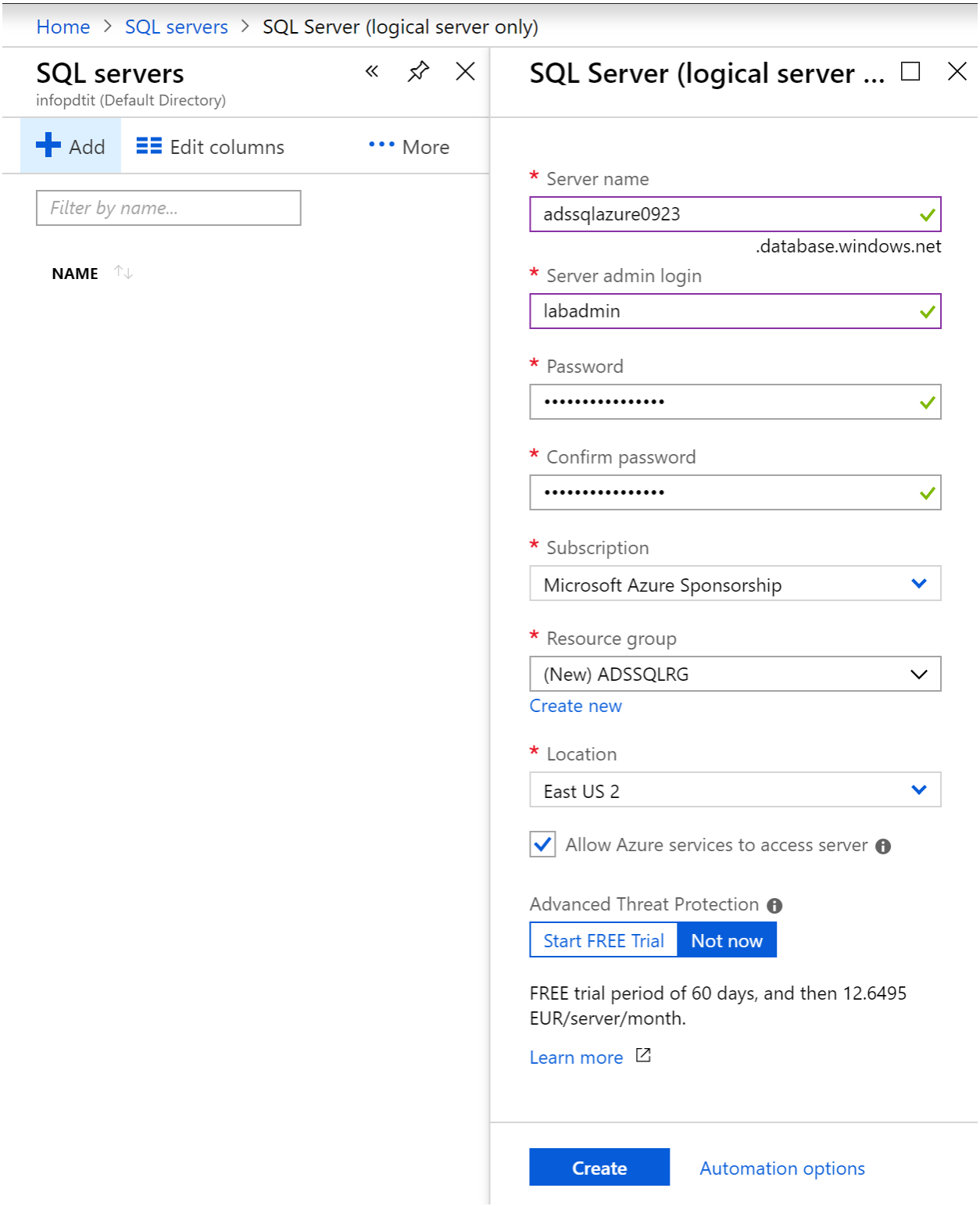
1. From within the **Azure Portal**, Go to "**All Services**", and enter "**SQL Server**" in the search field. From the list of results, select **SQL Servers**.




2. **Click** "Create a new SQL Server", or **Click** the "**+Add**" button in the top menu. This launches the SQL Server (logical server) deployment blade.

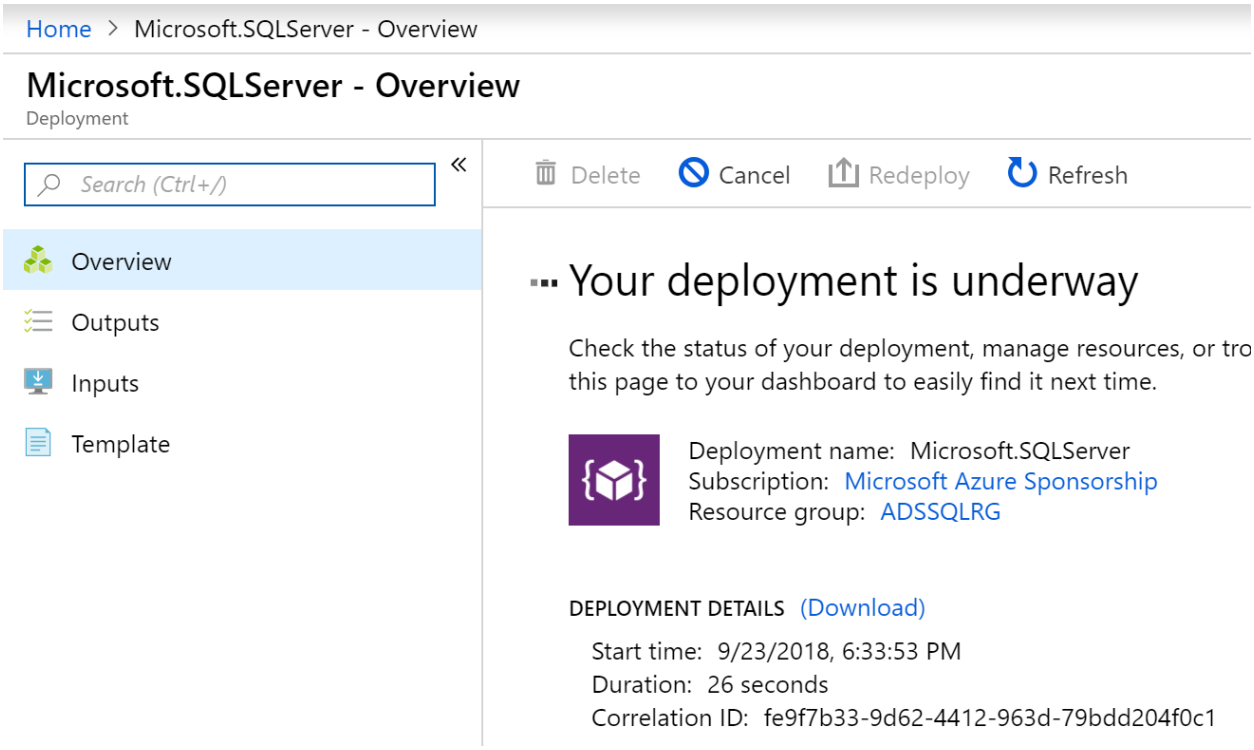



3. Complete the different deployment settings as follows:
   - Server Name: [suffix]sqlazure[date] e.g. adssqlazure0923 (capitals are not allowed)
   - Server admin login: labadmin
   - Password: L@BadminPa55w.rd / confirm password: L@BadminPa55w.rd
   - Subscription: your Azure subscription
   - Resource Group: Create New / [suffix]SQLRG
   - Location: Azure location close to you
   - Allow Azure Services to access server: checked
   - Advanced Threat Protection: Not Now




4. Confirm the creation of the SQL Azure Server by pressing the Create button.

5. **Wait** for the deployment to complete. You can follow the deployment from the notification area or selecting deployment from the new Resource Group you defined.



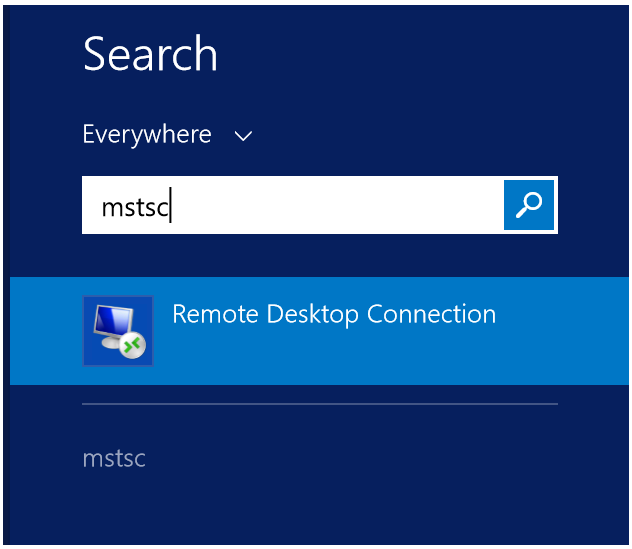

6. Once the SQL Azure server has been deployed successfully, continue to Task 2.

## Task 2: Performing a SQL database migration from a SQL Virtual Machine to SQL Azure, using SQL Management Studio.

In this task, you perform a SQL database migration from within a SQL Virtual Machine to SQL Azure. This approach is known as a lift & shift database migration, since no structure or data will be changed during the actual migration.

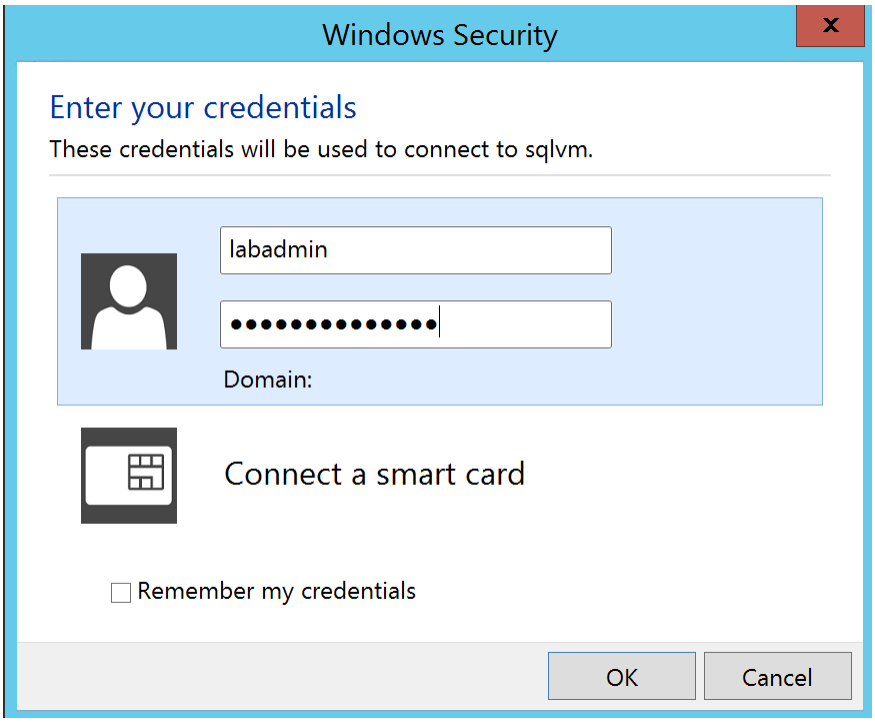
1. **Open an RDP session** to the **WebVM Virtual Machine** (using the same steps as described in the previous lab). This is required, since the SQLVM has no public IP-address.

2. Once you are logged on to the WebVM RDP session, **start a new RDP session to the SQLVM**: from the **WebVM desktop**, press the **Start button**. From the **Start screen**, start typing "**mstsc**", which resolves Remote Desktop Connection.




3. Click on **Remote Desktop Connection**.

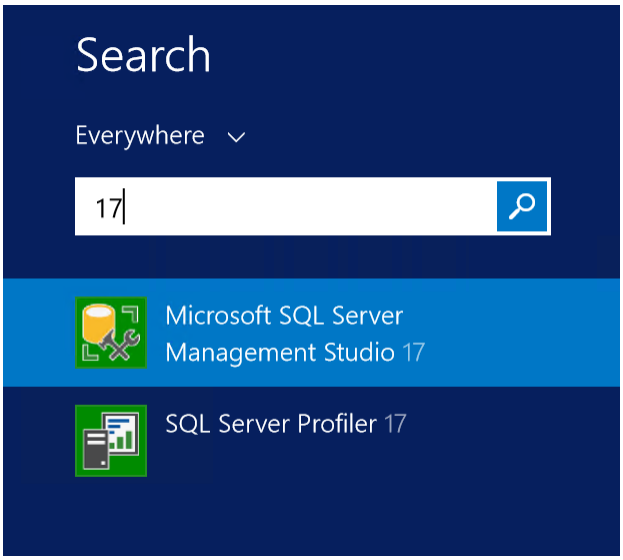
4. As server name, type "**SQLVM**". (Since both Virtual Machines are in the same Azure Virtual Network and subnet, the netbios name resolution works, relying on Azure DNS). **Press connect.**




5. Provide the local admin credentials of the SQLVM Virtual Machine:
   - labadmin
   - L@BadminPa55w.rd

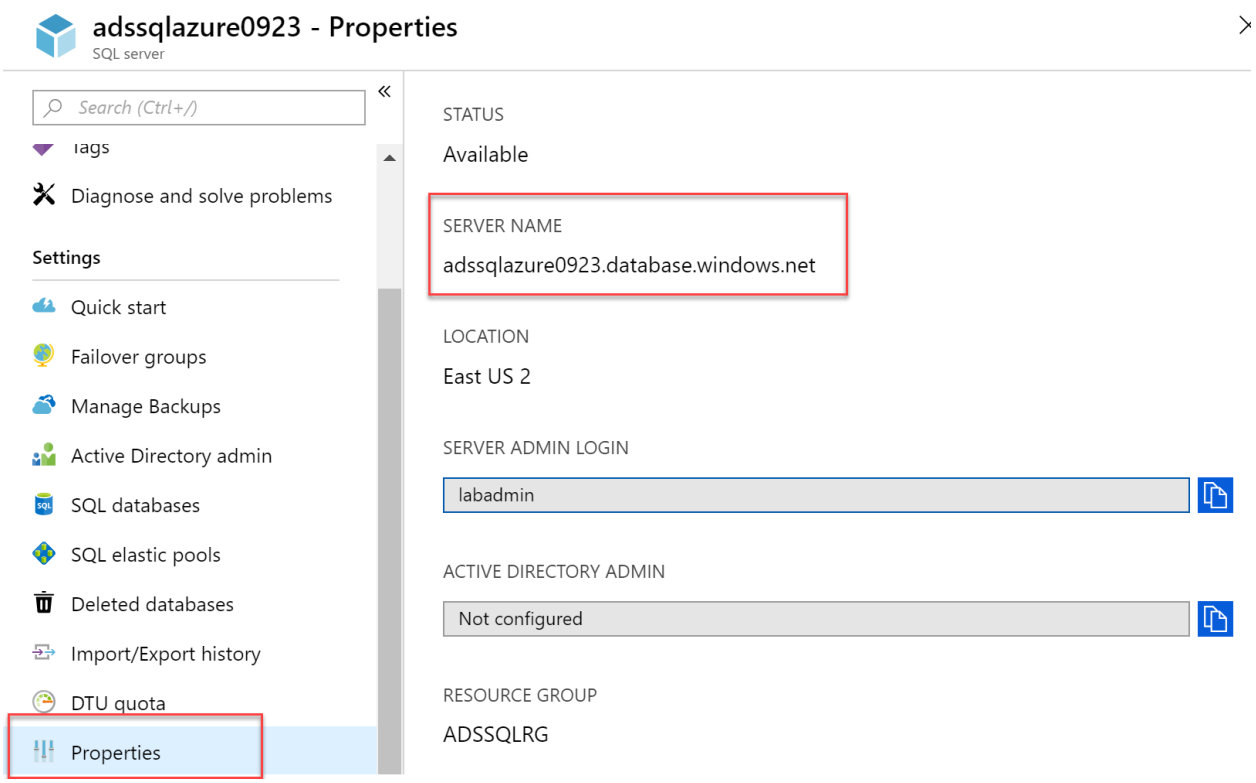
and confirm with **OK**.

6. Once you **are logged on** to the **SQL Server Virtual Machine** (notice the SQL Getting Started shortcut on the desktop), press the **Start button**. **Start typing "17"**; this will resolve several management tools available on the server. Notice **Microsoft SQL Server Management Studio 17**.

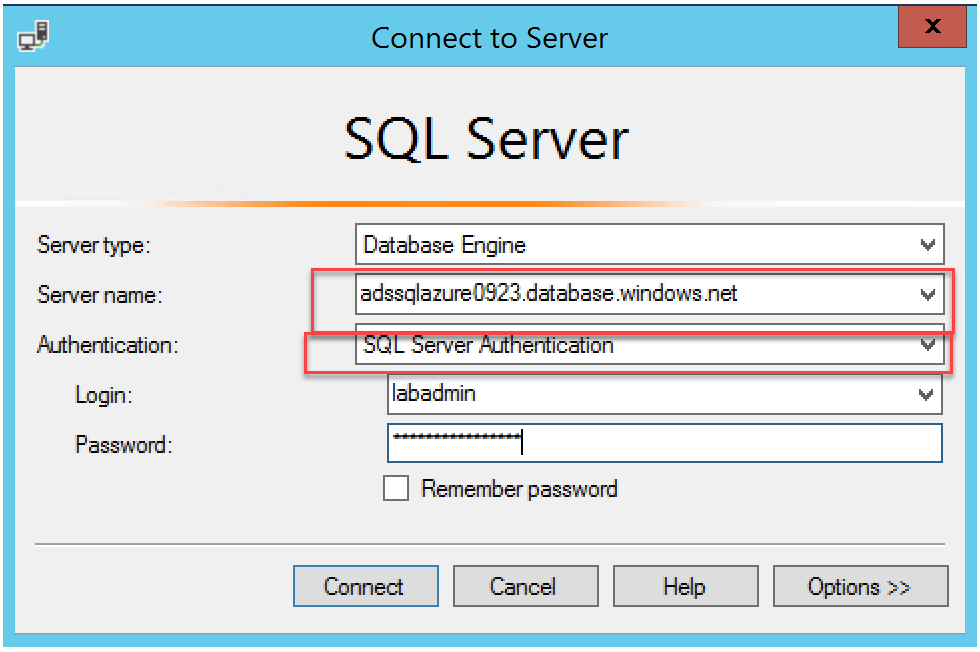



7. Select it to start the **SQL Server Management Studio 17 console.**

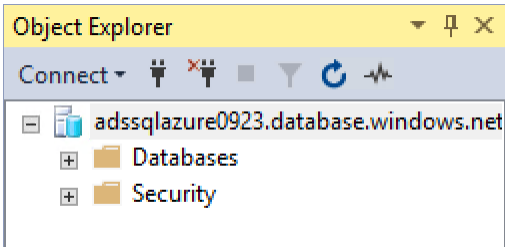
8. Once opened, you are asked for **server connection information**. This is where you provide the SQL Azure name. You can find this in the **Azure Portal**, by browsing to your **SQL Azure instance**, and selecting its **properties**



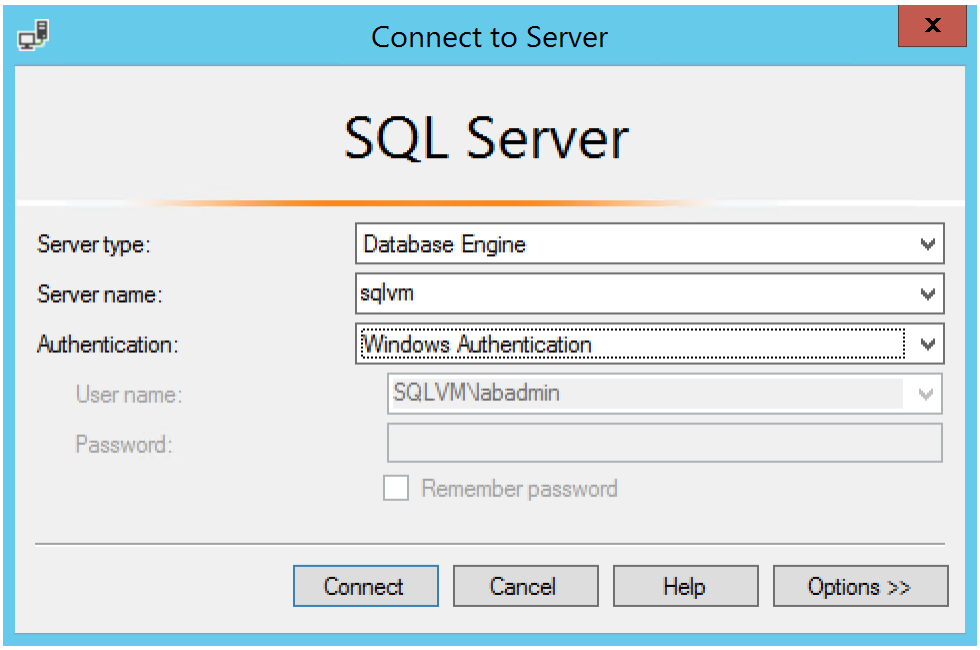

9. Copy the SERVER NAME into the Server Name field of the SQL connection popup; in the Authentication field, change to SQL Server Authentication. Provide the Login and Password of the SQL Azure instance account you provided during the deployment of this resource. (labadmin / L@BadminPa55w.rd would be the instructed ones)



10. Press Connect to log on to this SQL Server instance.

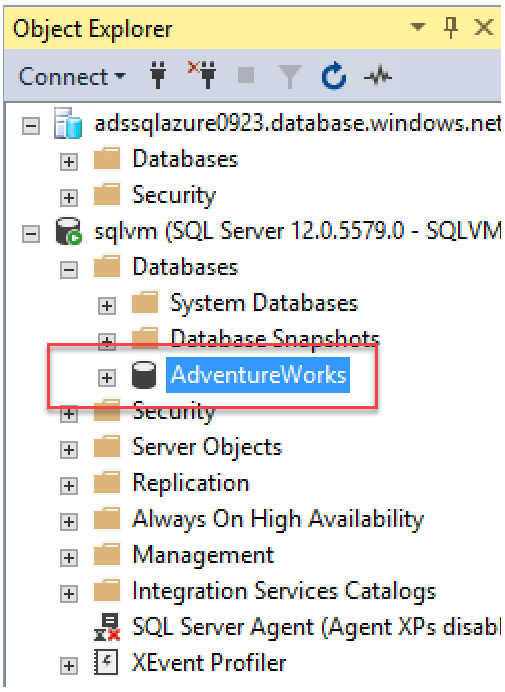



11. In order to have a connection to the SQLVM database instance, we need to add another connection. From the SQL Management Studio console, click File / Connect Object Explorer. In the Connect to server popup that appears, this time provide the server credentials from the SQLVM:
server name: sqlvm
authentication: Windows Authentication (this is possible since you are logged on to the SQL VM Virtual Machine already. Notice the admin and password field will become greyed out.

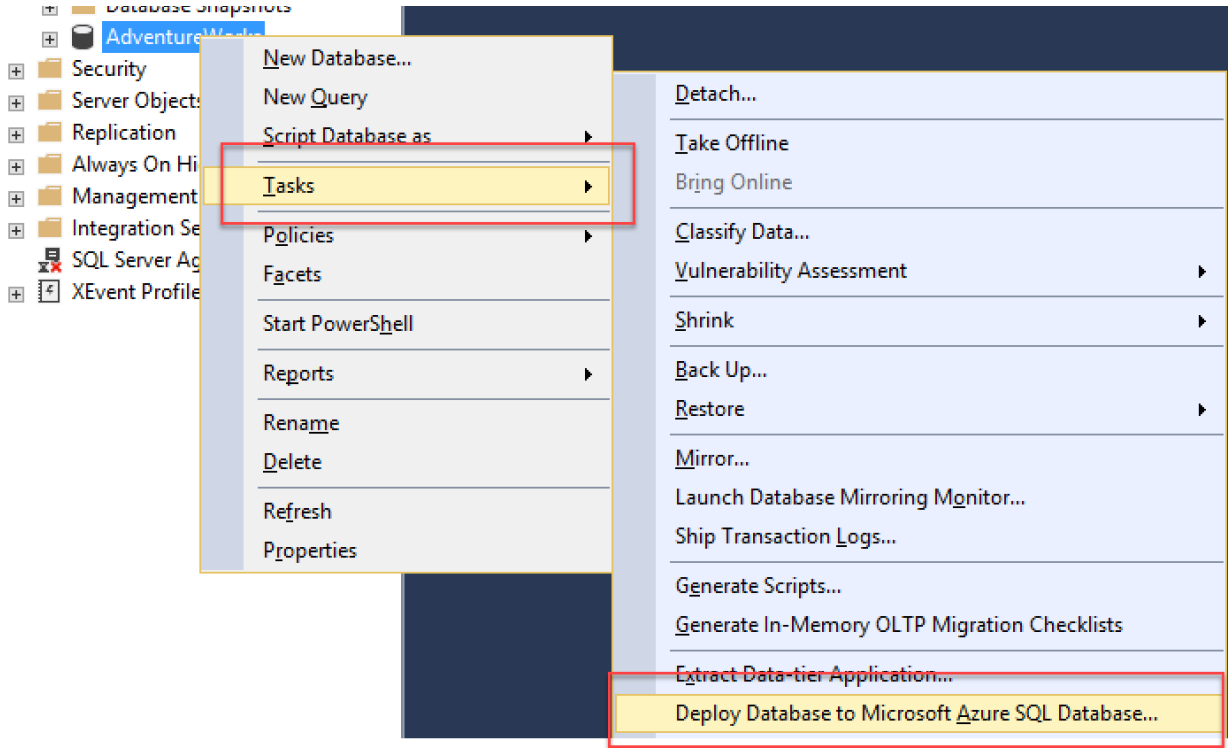



12. Press the Connect button.

13. The Object explorer shows a successful connection to both databases now. If you open the Databases level, you should see the AdventureWorks database.

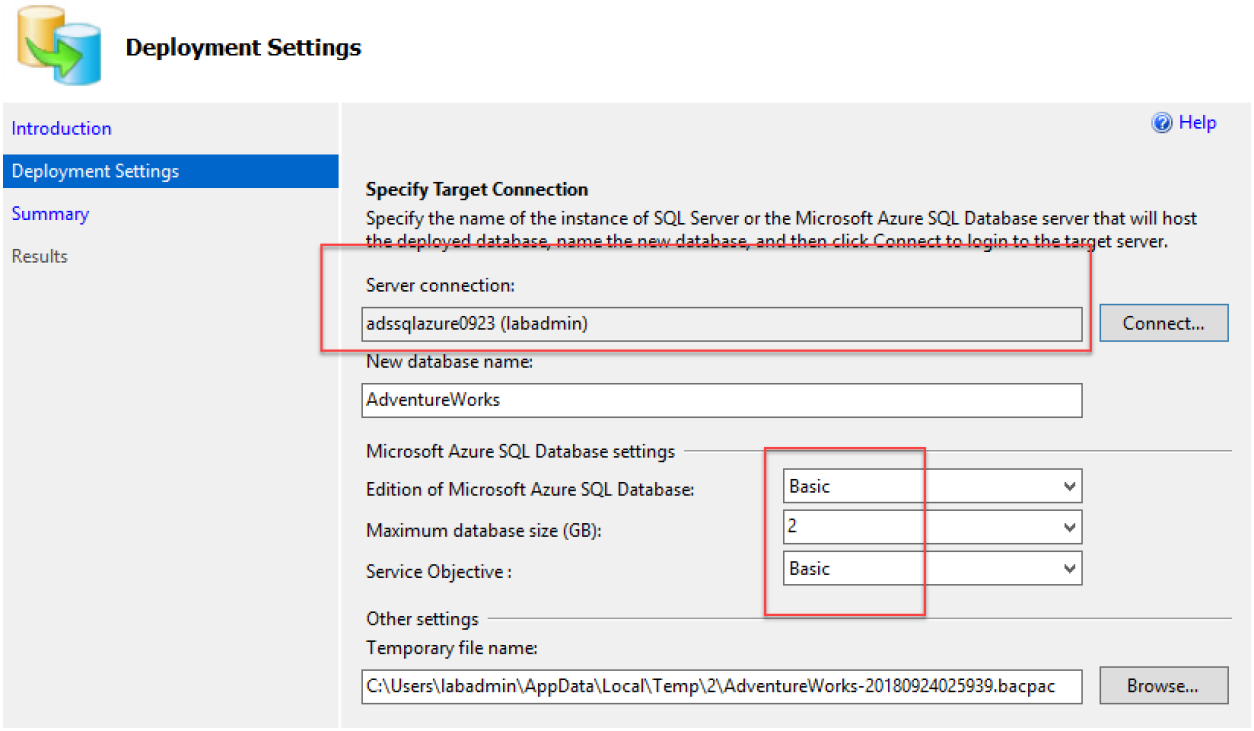



14. The next step is running the actual migration of the database. Therefore, **select** the AdventureWorks database on the local sqlvm, **right-click** it, select **Tasks,** and next, select Deploy Database to Micrsoft SQL Azure Database.

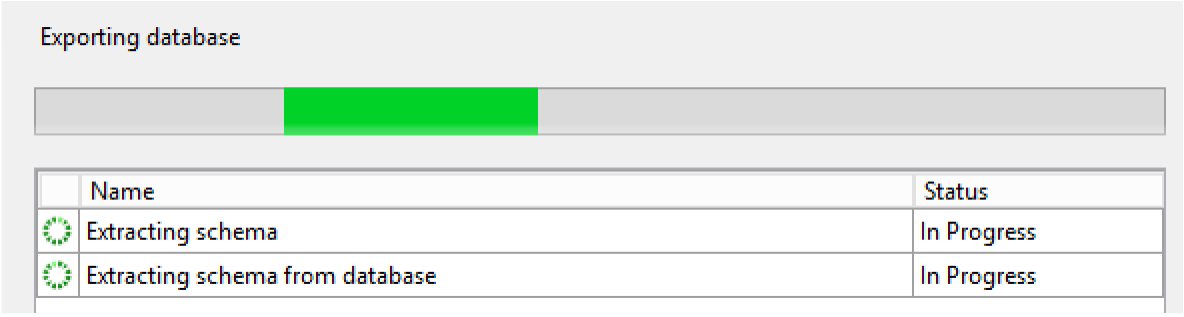


15. Press the Next button when you see the Introduction step showing up.

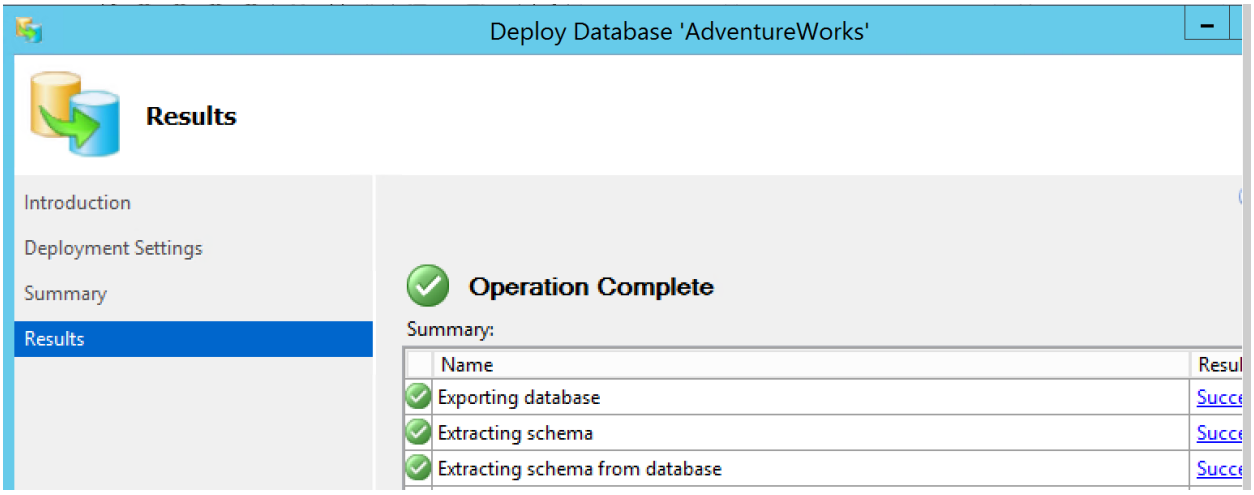
16. In the Deployment Settings, provide the Server Connection by pressing the Connect button. Provide the following details here:
    - Server Connection: <your sql server in Azure>
    - New Database name: AdventureWorks
    - Edition of Microsoft SQL Database: Basic
    - Max DB size: 2GB
    - Service Objective: Basic




17. Read through the settings in the summary step. Press the Finish button to start the actual move process.



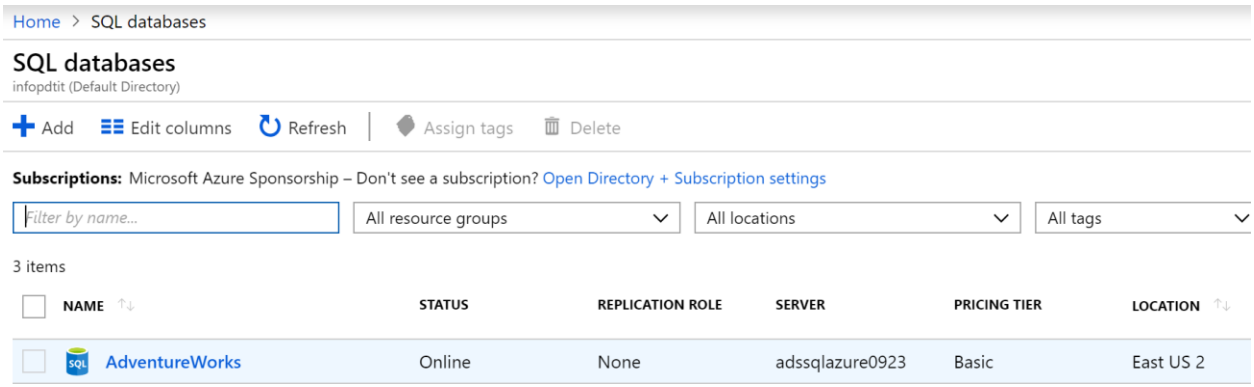
18. Wait for this process to complete – this could take about 10minutes.



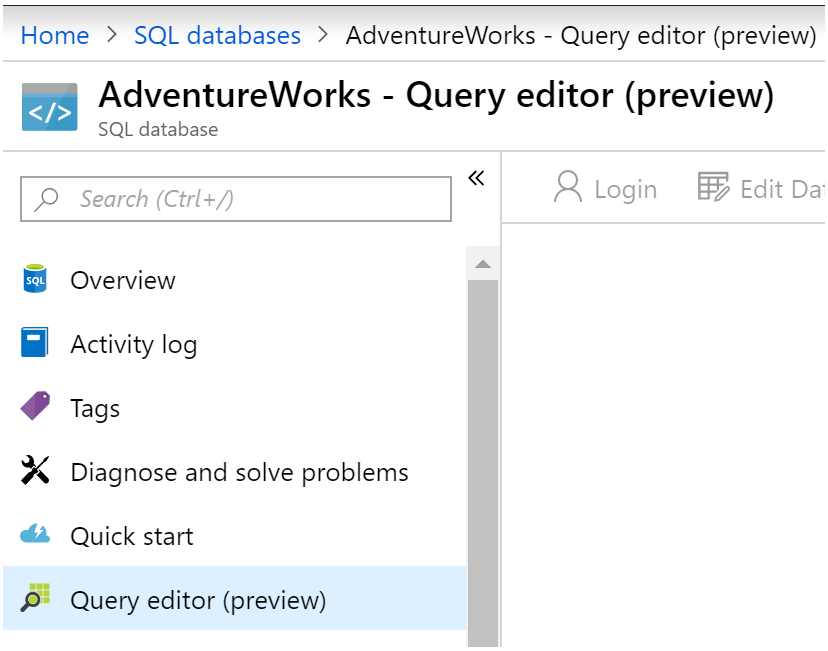

19. Once completed, close the migration window.

20. To validate the database is actually migrated successfully, one can go back to the Azure Portal, and use the Query editor, which is currently in preview, to check on the contents of the database and tables. From the Azure Portal, navigate to all services and select SQL databases.

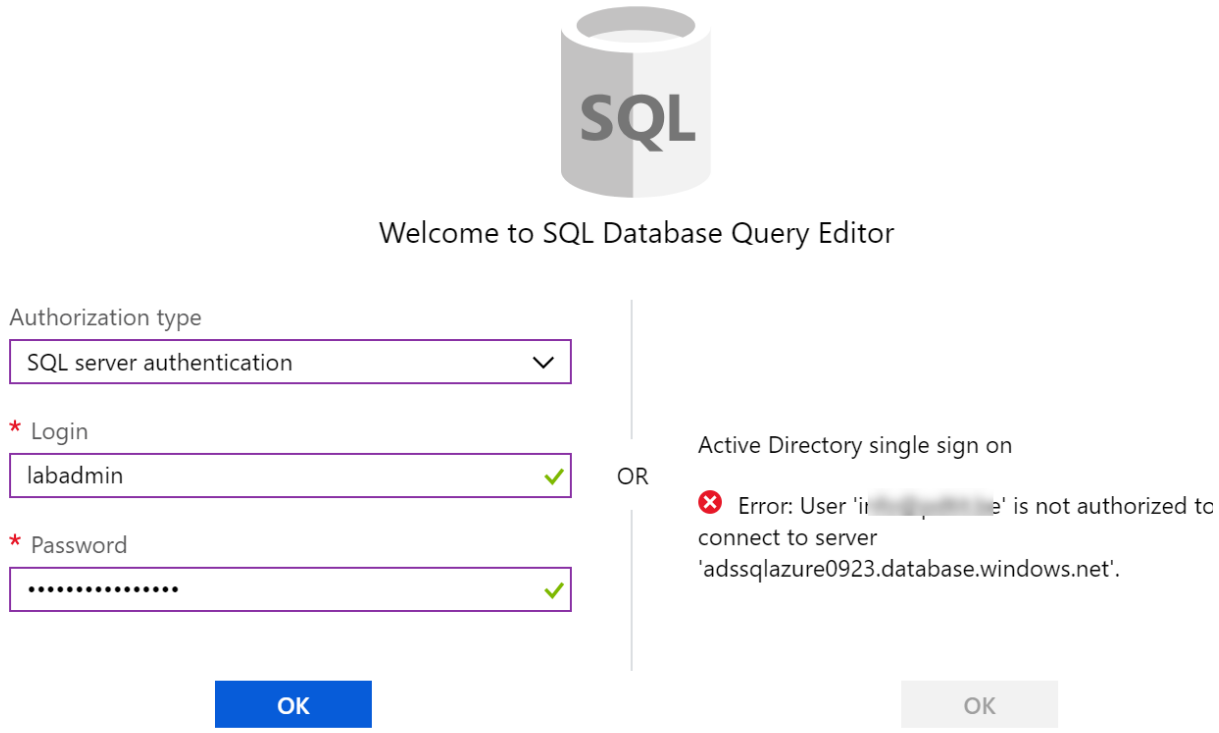
21. From the list of databases, select AdventureWorks.




22. From the Adventureworks detailed blade, select Query editor (preview)



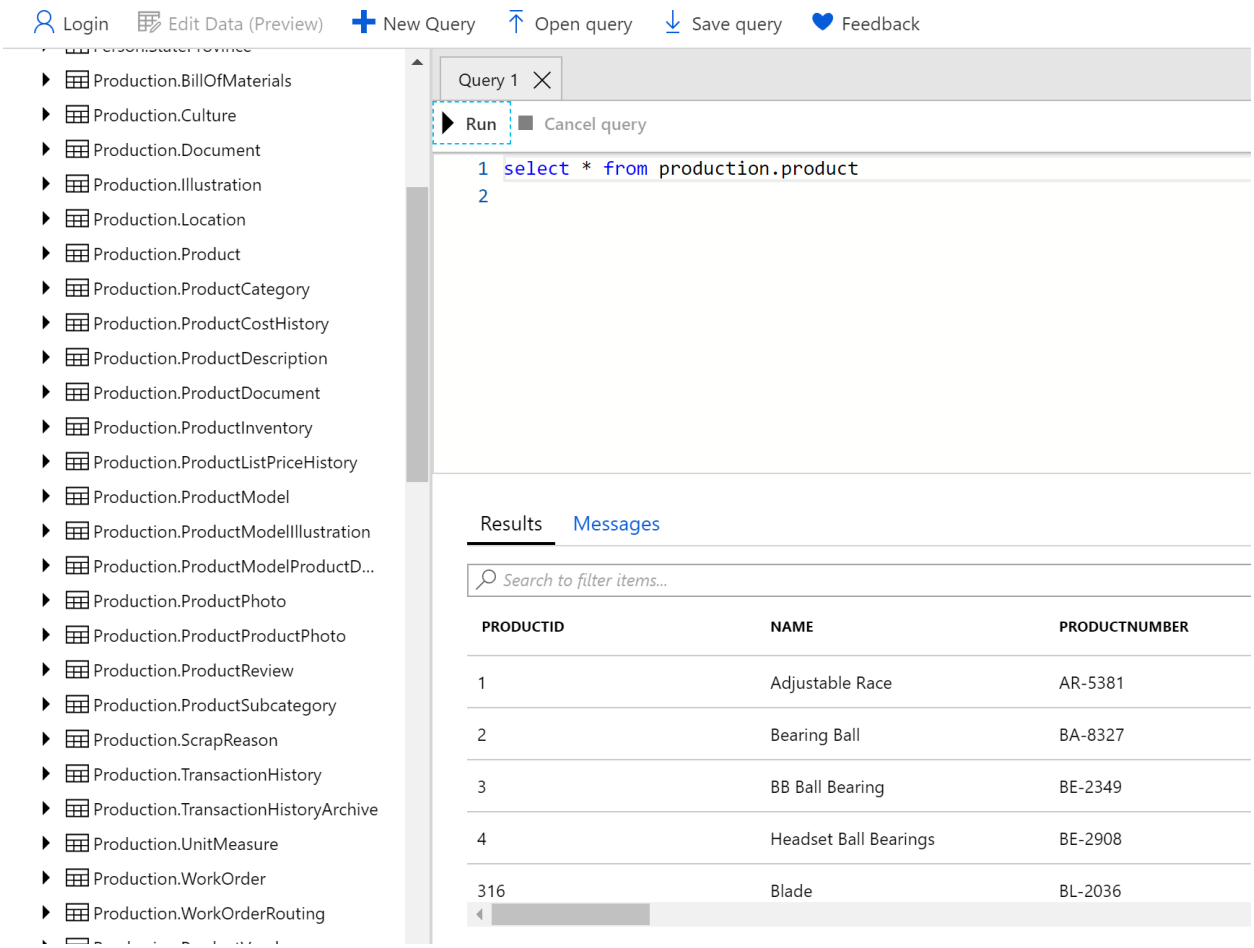

23. Notice you are prompted to provide your SQL database credentials, where your Azure admin credentials are not giving you access to the database content.



24. Provide the labadmin / L@BadminPa55w.rd credentials, and **press OK**. This opens the query editor window. In the Query editor to the right, enter the following SQL query:

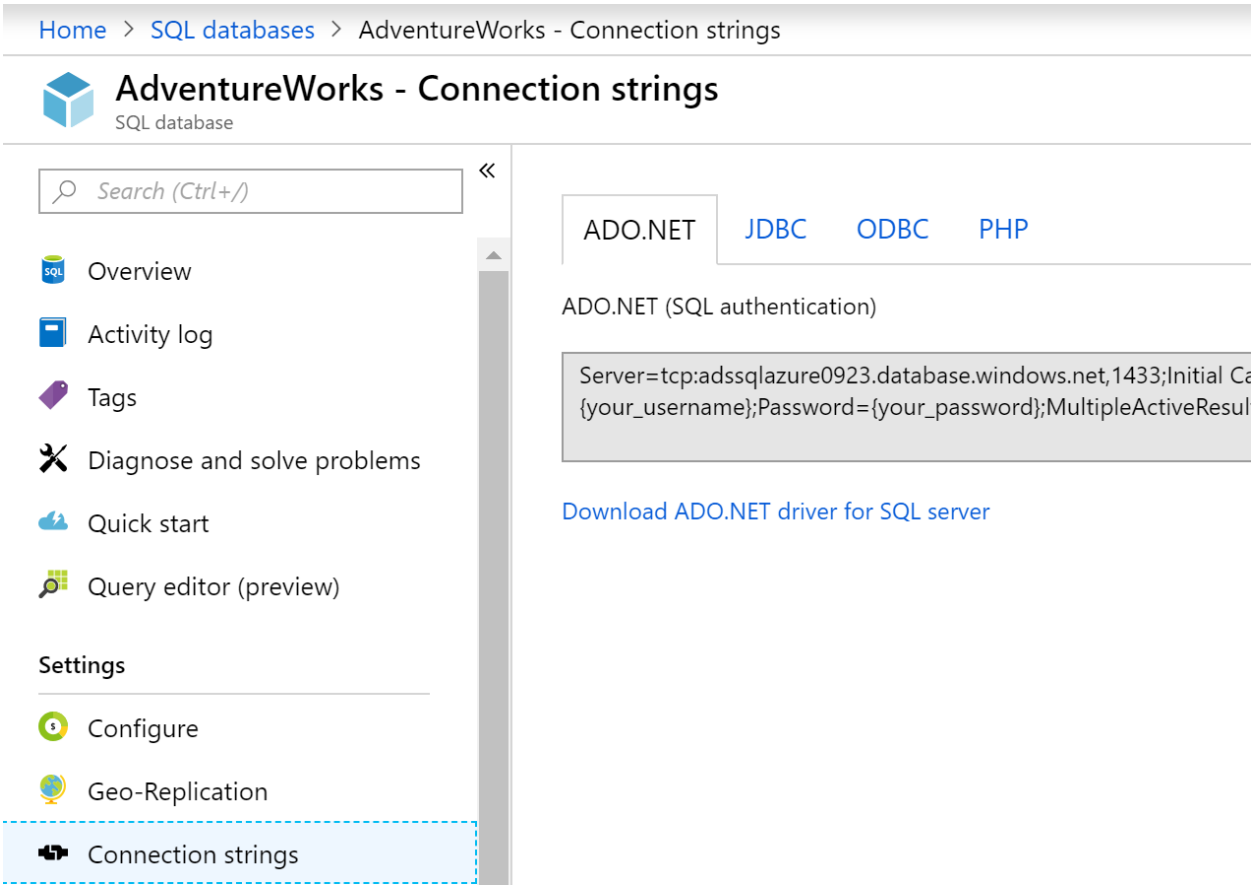
```
select * from Production.Product
```

and **press Run**. This will show you the full list of all Products we have in the database. Which confirms our migration from SQL Management Studio ran successful.




## Task 3: Defining a hybrid connection from a Web VM to an Azure SQL database

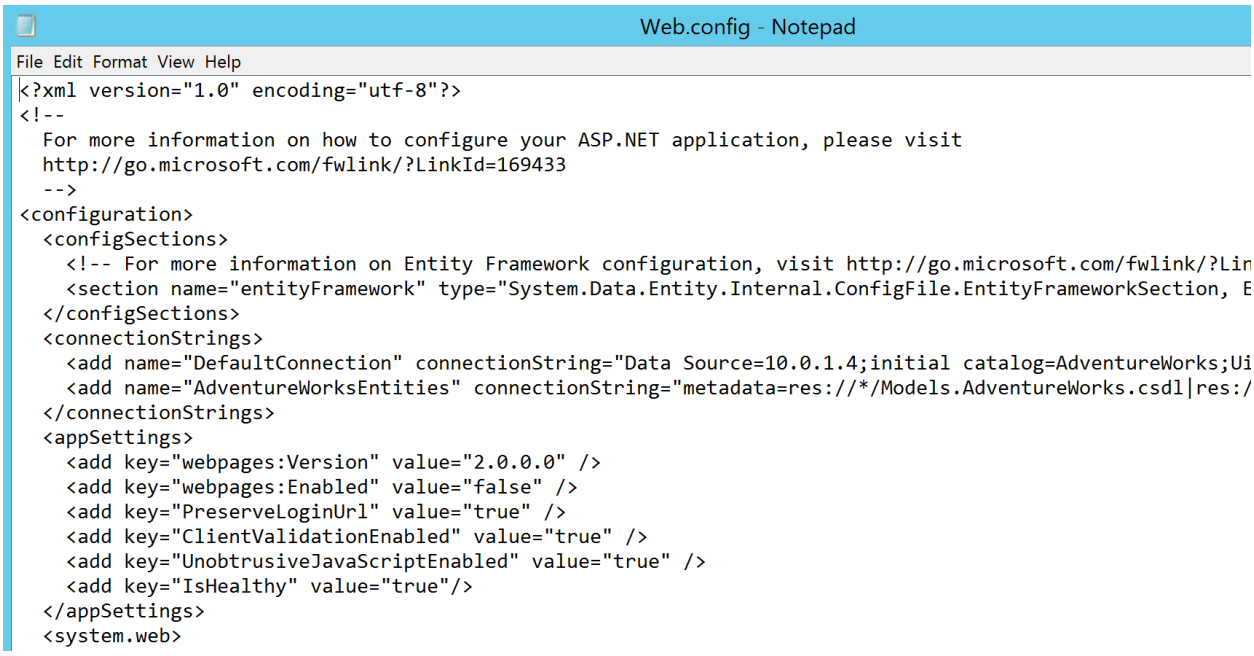
1. To complete our hybrid cloud migration, we will now update the connection string settings in the web.config file of our WebVM web site application. This information can be retrieved from the SQL database settings in the Azure Portal. **From within the SQL database detailed blade**, browse to **Connection String** under the **settings** section.




2. Leave this information on screen, as you will need to copy parts of the ADO.NET connection string information into the Web server's web.config file.

3. Go back to the WebVM Virtual Machine Remote Desktop session (or open it again when you already closed the WebVM RDP session)

4. Browse to the IIS web server folder that has the web application content: c:\inetpub\wwwroot\ and open the file **web.config** with Notepad.



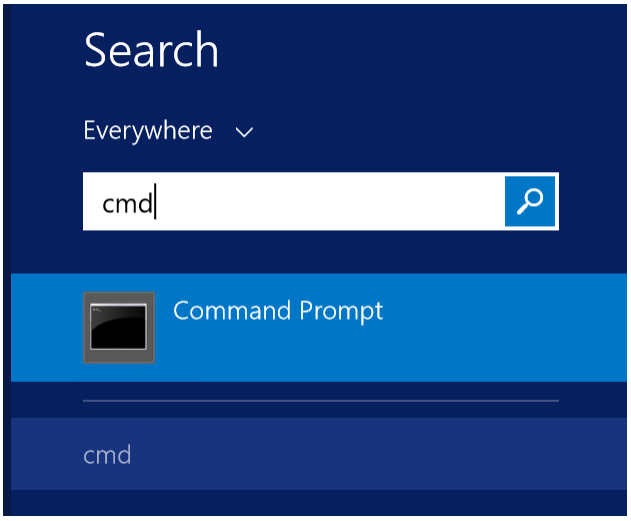

5. Go to the section that starts with **<connectionStrings>**. **Replace the** following settings with the parameters from the Connection String information in the Azure Portal, in both lines starting with <add name:
   **- Data Source=10.0.1.4 => change** the 10.0.1.4 with your SQL Server name e.g. adssqlazure0923.database.windows.net in our example
- **Uid=sa => change** the sa account to labadmin

```
<connectionStrings>
<add name="DefaultConnection" connectionString="Data Source=adssqlazure0923.database.windows.net;
initial catalog=AdventureWorks;Uid=labadmin;Password=L@BadminPa55w.rd;MultipleActiveResultSets=True"
providerName="System.Data.SqlClient" />
<add name="AdventureWorksEntities" connectionString="metadata=res://*/Models.AdventureWorks.csdl
|res://*/Models.AdventureWorks.ssdl |res://*/Models.AdventureWorks.msl;provider=System.Data.SqlClient;
provider connection string=&quot;data source=adssqlazure0923.database.windows.net;initial
catalog=AdventureWorks;Uid=labadmin;Password=L@BadminPa55w.rd;multipleactiveresultsets=True;App=Entity
Framework&quot;" providerName="System.Data.EntityClient" />
  </connectionStrings>
```

6. **Save** the changes to the web.config file.

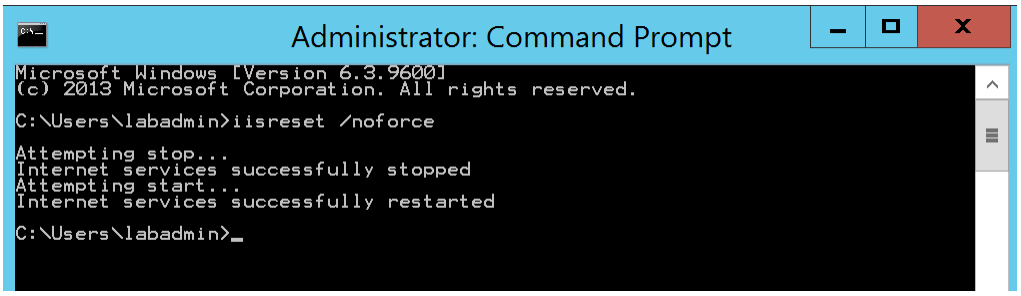
7. As we will need these connection strings in a later lab again, copy these updated lines (everything between <connectionStrings> and </connectionStrings>) to a new Wordpad text document, preferably saving it on the lab-jumpVM for later retrieval. Easiest to achieve this is Copy/Paste from the WebVM to a new text document on the lab-jumpVM, and saving it on the lab-JumpVM.

8. **From the Start Screen** on the WebVM, **open a command prompt, by typing "CMD"**.



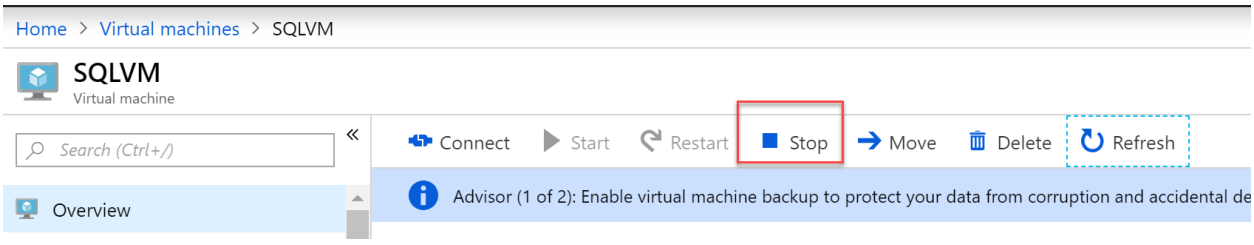

9. In the command prompt, run the following command, to restart the IIS Web Server service.

```
iisreset /noforce
```



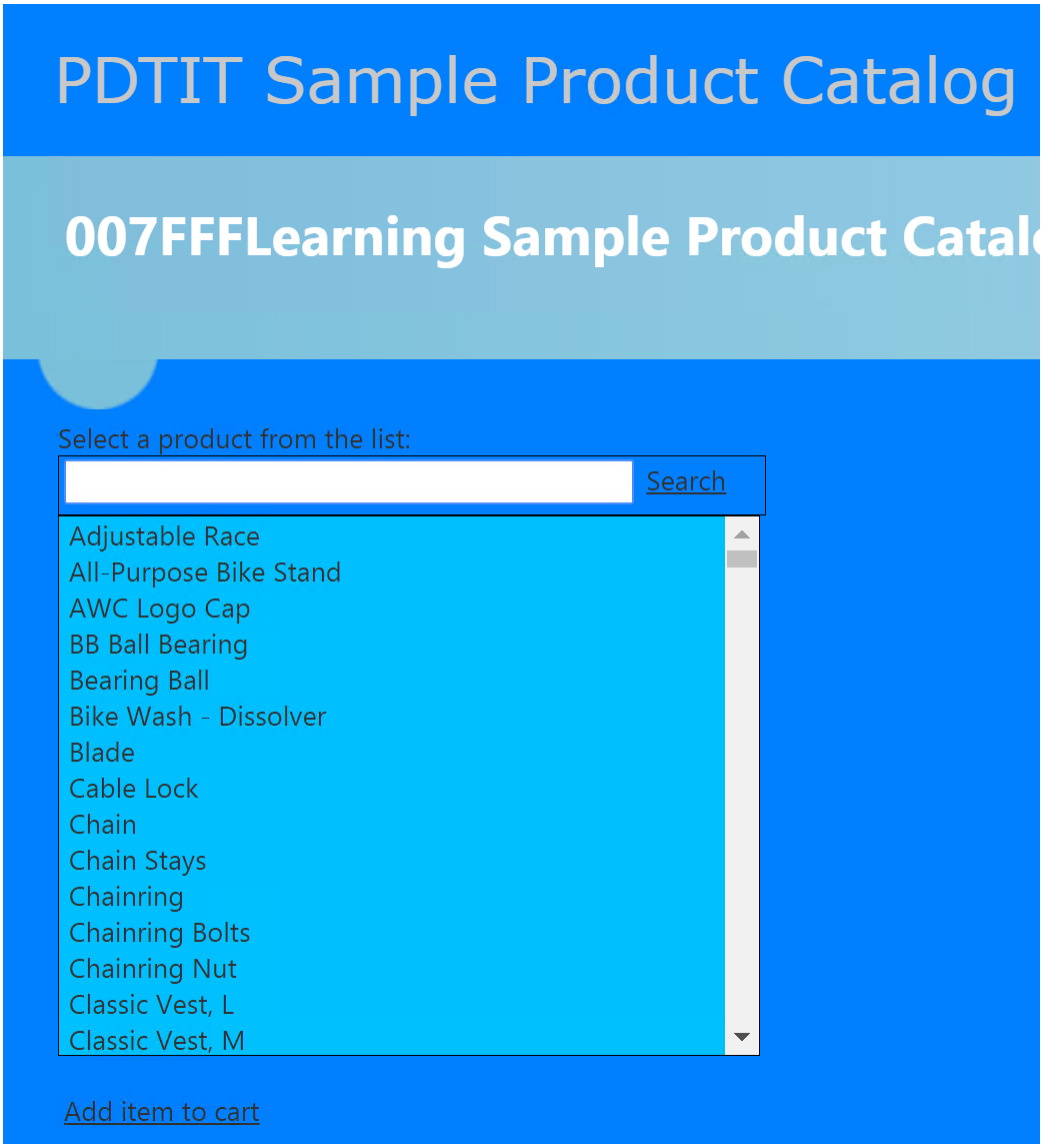

10. To proof that the web application is now connecting to the Azure SQL database, let's shutdown the SQLVM. From the Azure Portal, navigate to Virtual Machines, and click on the SQLVM Virtual Machine.

11. From the SQLVM detailed blade, press the STOP button in the top menu. Wait for the notification message, telling you the VM has shutdown.



12. To test if the web application is now connecting to the Azure SQL database, browse to the web site from within the WebVM's browser, connecting to localhost.

13. The website should load successfully and showing you the product catalog list.




14. This completes this lab.

## Summary

In this lab, you learned how to deploy a SQL Azure server resource, as well as how to migrate a SQL database using SQL Server Management Studio 17. You updated the IIS web server web.config file and validated the web application is now running in a hybrid setup.