

# Azure Developer Series

## Migrating a legacy ASP.NET 2-tier application to Azure using Container Services

Hands-On-Labs step-by-step guides

Prepared by:

Peter De Tender

CEO and Lead Technical Trainer  
PDTIT and 007FFFlearning.com

@pdtit

@007FFFlearning

Version: April 2019

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2018 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.



# Contents

Abstract and Learning Objectives .....	4
Hands-On-Lab Scenario .....	5
Requirements .....	5
Naming Conventions:.....	5
Azure Subscription:.....	5
Other requirements: .....	5
Alternative Approach: .....	6
Final Remarks:.....	6
Lab 2: Migrating a SQL Server database to SQL Azure .....	7
What you will learn .....	7
Time estimate .....	7
Task 1: Deploying a new SQL Azure Server instance .....	7
Task 2: Performing a SQL database migration from a SQL Virtual Machine to SQL Azure, using SQL Management Studio.....	9
Task 3: Defining a hybrid connection from a Web VM to an Azure SQL database.....	17
Summary.....	21
Optional Lab on Azure Database Migration Assistant .....	22
Task 1: Installing Azure Database Migration Assistant .....	22

# Migrating a legacy ASP.NET 2-tiered application to Azure using Container Services - Hands-On-Labs step-by-step

## Abstract and Learning Objectives

This workshop enables anyone to learn, understand and build a Proof of Concept, in performing a multi-tiered legacy ASP.NET web application using Microsoft SQL Server database, platform migration to Azure public cloud, leveraging on different Azure Platform Azure A Service (PaaS) and Azure Container Services.

After an introductory module on cloud app migration strategies and patterns, students get introduced to the basics of automating Azure resources deployments using Visual Studio and Azure Resource Manager (ARM) templates. Next, attendees will learn about Microsoft SQL database migration to SQL Azure PaaS, as well as deploying and migrating Azure Web Apps.

After these foundational platform components, the workshop will totally focus on the core concepts and advantages of using containers for running web apps, based on Docker, Azure Container Registry (ACR), Azure Container Instance (ACI), as well as how to enable container cloud-scale using Azure Container Services (ACS) with Kubernetes and Azure Kubernetes Service (AKS).

The focus of the workshop is having a Hands-On-Labs experience, by going through the following exercises and tasks:

- Deploying a 2-tier Azure Virtual Machine (Webserver and SQL database Server) using ARM-template automation with Visual Studio 2017;
- Migrating a legacy SQL 2012 database to Azure SQL PaaS (Lift & Shift);
- Migrating a legacy ASP.NET web application to Azure Web Apps (Lift & Shift);
- Containerizing a legacy ASP.NET web application using Docker;
- Running Azure Container Instance (ACI) from an Azure Container Registry (ACR) image;
- Deploy and run Azure Container Services (ACS) with Kubernetes;
- Deploy and run Azure Kubernetes Services (AKS);
- Managing and Monitoring Azure Container Services (ACS) and Azure Kubernetes Services (AKS);

## Hands-On-Lab Scenario

The baseline of the hands-on-lab scenario is starting from an 8-year-old legacy ASP.NET application, developed around 2010, currently offering a product catalog browsing web page, pulling the product catalog list from a legacy Microsoft SQL Server 2012 database, running on dedicated Windows Server 2012 R2 Virtual Machines. (This could be seen as a subset of a larger application landscape within your organization, think of manufacturing database information, HR solutions, e-commerce platforms,... and alike). Imagine this application is running in our on-premises datacenter, where you want to perform an “application digital migration” to Azure Public cloud. You will use several Azure cloud-native services and solutions, ranging from Virtual Machines, Platform Services and different Container Solutions on Azure.

## Requirements

### Naming Conventions:

IMPORTANT: Most Azure resources require unique names. Throughout these steps you will see the word “[SUFFIX]” as part of resource names. You should replace this with your initials, guaranteeing those resources get uniquely named.

### Azure Subscription:

Participants need a “pay-as-you-go”, MSDN or other paid Azure subscription

- a) **Azure Trial subscriptions won't work**
- b) In one of the Azure Container Services tasks, you are required to create an Azure AD Service Principal, which typically requires an Azure subscription owner to log in to create this object. If you don't have the owner right in your Azure subscription, you could ask another person to execute this step for you.
- c) The Azure subscription must allow you to run enough cores, used by the baseline Virtual Machines, but also later on in the tasks when deploying the Azure Container Services, where ACS agent and master machines are getting set up. If you follow the instructions as written out in the lab guide, you need 12 cores.
- d) If you run this lab setup in your personal or corporate Azure payable subscription, using the configuration as described in the lab guide, the estimated Azure consumption costs for running the setups during the 2 days of the workshop is \$20.

### Other requirements:

Participants need a local client machine, running a recent Operating System, allowing them to:

- browse to <https://portal.azure.com> from a most-recent browser;
- establish a secured Remote Desktop (RDP) session to a lab-jumpVM running Windows Server 2016;

## Alternative Approach:

Where the lab scenario assumes all exercises will be performed from within the lab-jumpVM, (since several tools will be installed on the lab-jumpVM or are already installed by default), participants could also execute (most, if not all...) steps from their local client machine.

The following tools are being used throughout the lab exercises:

- Visual Studio 2017 community edition (updated to latest version)
- Docker for Windows (updated to latest version)
- Azure CLI 2.0 (updated to latest version)
- Kubernetes CLI (updated to latest version)

Make sure you have these tools installed prior to the workshop, if you are not using the lab-jumpVM. You should also have full administrator rights on your machine to execute certain steps within using these tools.

## Final Remarks:

**VERY IMPORTANT:** You should be typing all of the commands as they appear in the guide, except where explicitly stated in this document. Do not try to copy and paste from Word to your command windows or other documents where you are instructed to enter information shown in this document. There can be issues with Copy and Paste from Word or PDF that result in errors, execution of instructions, or creation of file content.

**IMPORTANT:** Most Azure resources require unique names. Throughout these steps you will see the word "[SUFFIX]" as part of resource names. You should replace this with your initials, guaranteeing those resources get uniquely named.

## Lab 2: Migrating a SQL Server database to SQL Azure

### What you will learn

In this lab, you perform a migration from a SQL 2014 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;

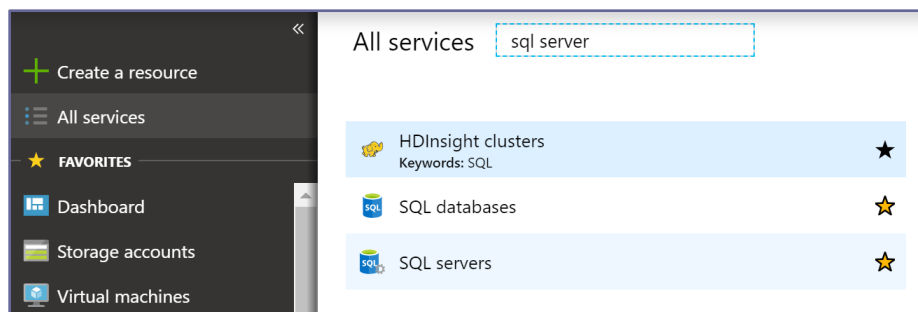
### Time estimate

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

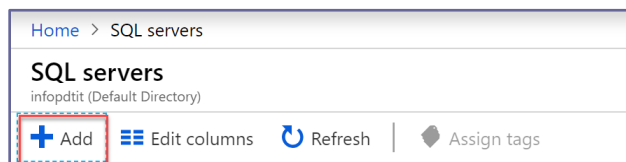
### Task 1: Deploying a new SQL Azure Server instance

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

Home > SQL servers > SQL Server (logical server only)

**SQL servers**  
infopdtit (Default Directory)

+ Add Edit columns More

Filter by name...

NAME ↑↓

**SQL Server (logical server ...**

\* Server name  
adssqlazure0923 ✓  
.database.windows.net

\* Server admin login  
labadmin ✓

\* Password  
..... ✓

\* Confirm password  
..... ✓

\* Subscription  
Microsoft Azure Sponsorship ▼

\* Resource group  
(New) ADSSQLRG ▼  
[Create new](#)

\* Location  
East US 2 ▼

☒ Allow Azure services to access server ⓘ

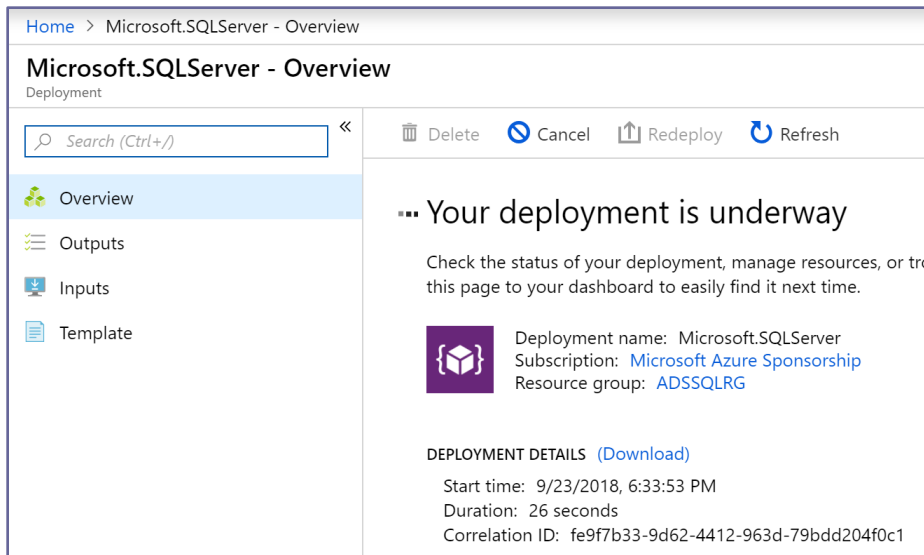
Advanced Threat Protection ⓘ  
[Start FREE Trial](#) [Not now](#)

FREE trial period of 60 days, and then 12.6495 EUR/server/month.  
[Learn more](#) ↗

[Create](#) [Automation options](#)

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.



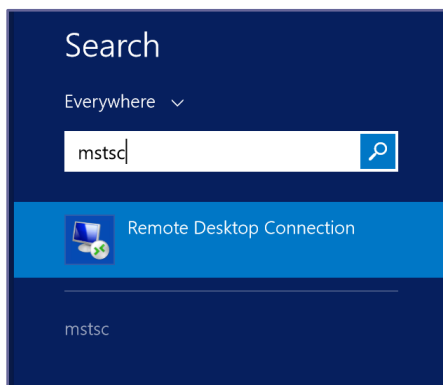


6. Once the SQL Azure server has been deployed successfully, continue to Task 2.

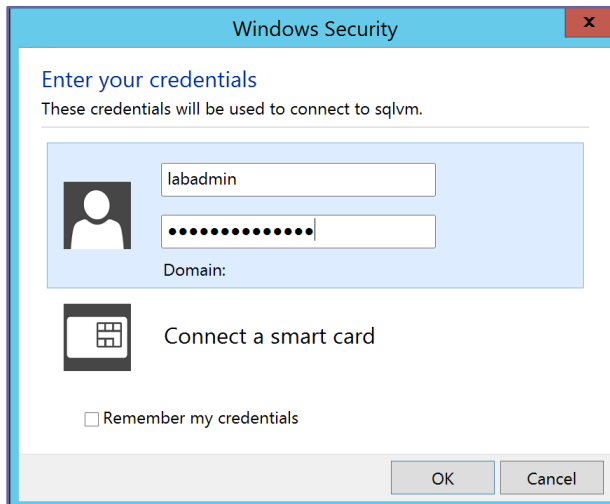
## Task 2: Performing a SQL database migration from a SQL Virtual Machine to SQL Azure, using SQL Management Studio.

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.

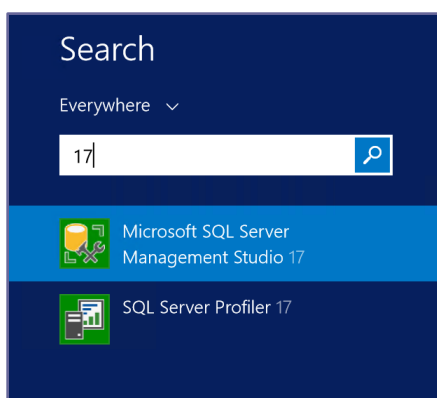
1. **Open an RDP session to the WebVM Virtual Machine** (using the same steps as described in the previous lab). This is required, since the SQLVM has no public IP-address.
2. Once you are logged on to the WebVM RDP session, **start a new RDP session to the SQLVM**: from the **WebVM desktop**, press the **Start button**. From the **Start screen**, start typing "mstsc", which resolves Remote Desktop Connection.



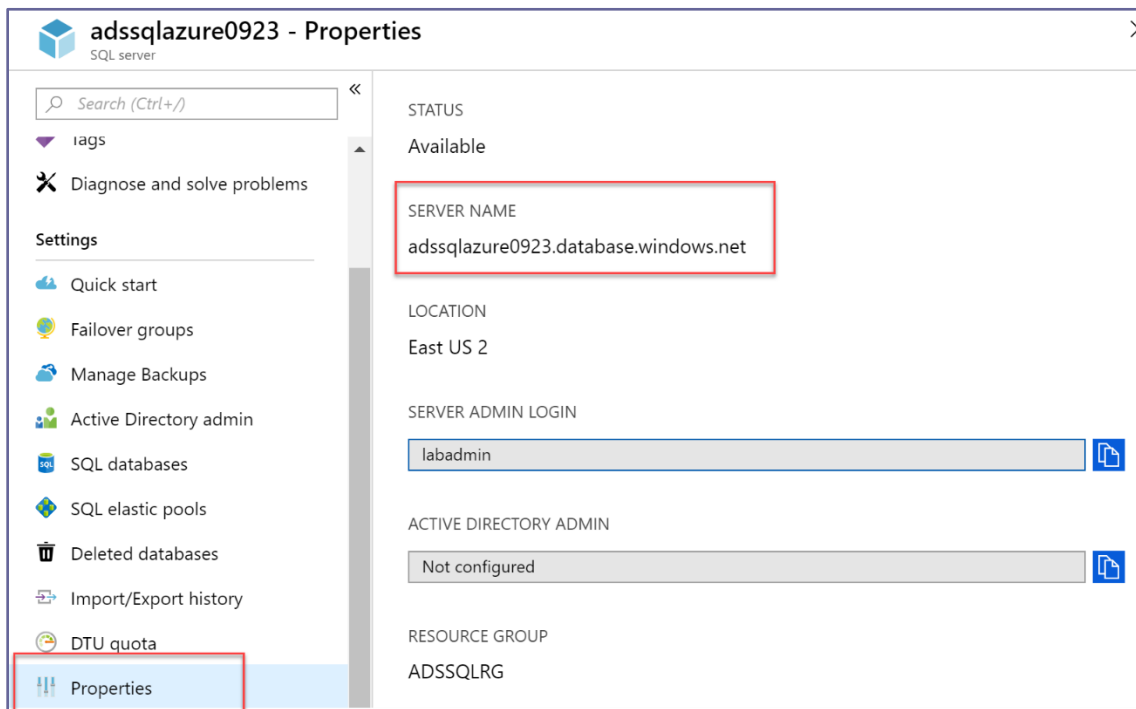
3. Click on **Remote Desktop Connection**.
4. As server name, type **"SQLVM"**. (Since both Virtual Machines are in the same Azure Virtual Network and subnet, the netbios name resolution works, relying on Azure DNS). **Press connect**.



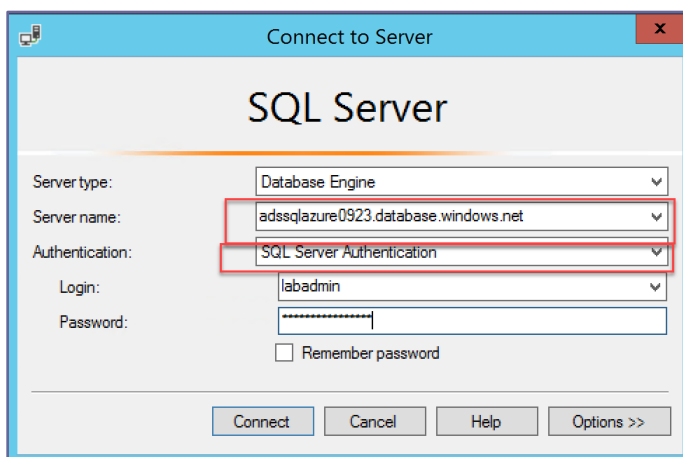
5. Provide the local admin credentials of the SQLVM Virtual Machine:
  - labadmin
  - [L@BadminPa55w.rd](#)and confirm with **OK**.
6. Once you **are logged on** to the **SQL Server Virtual Machine** (notice the SQL Getting Started shortcut on the desktop), press the **Start button**. **Start typing "17"**; this will resolve several management tools available on the server. Notice **Microsoft SQL Server Management Studio 17**.



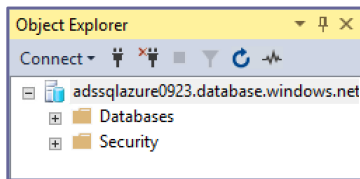
7. Select it to start the SQL Server Management Studio 17 console.
8. Once opened, you are asked for **server connection information**. This is where you provide the SQL Azure name. You can find this in the **Azure Portal**, by browsing to your **SQL Azure instance**, and selecting its **properties**



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)



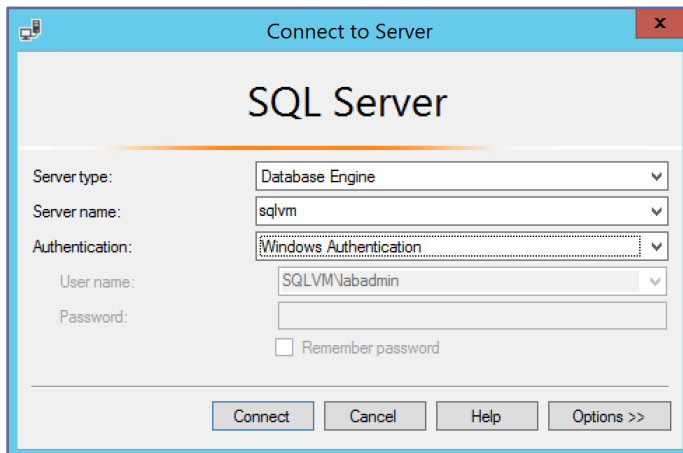
10. Press **Connect** to log on to this SQL Server instance.



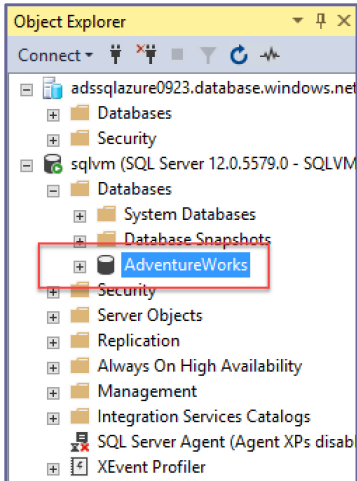
11. In order to have a connection to the SQLVM database instance, we need to add another connection. From the SQL Management Studio console, click **File / Connect Object Explorer**. In the **Connect to server** popup that appears, this time provide the server credentials from the SQLVM:

server name: sqlvm

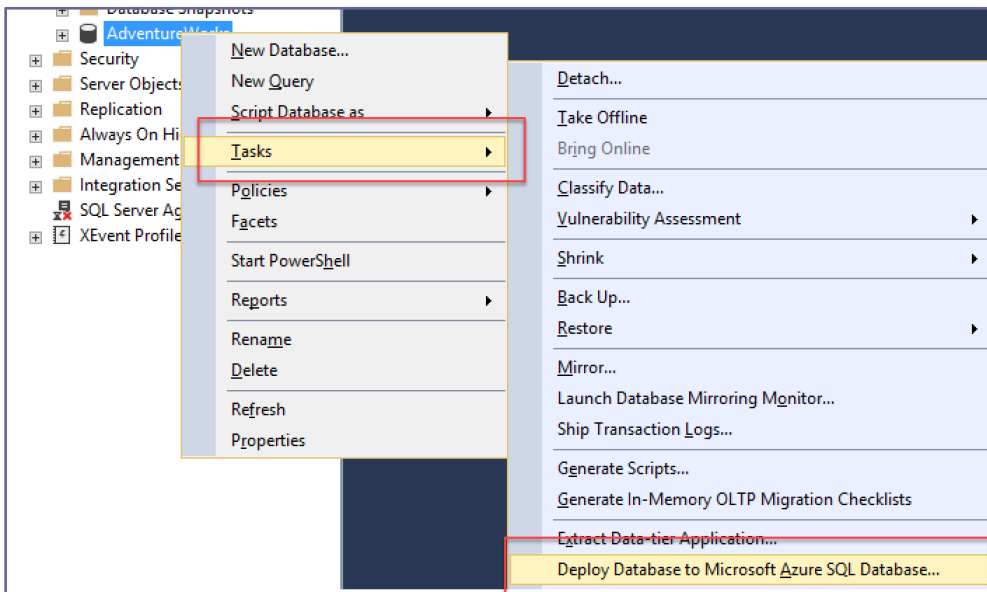
authentication: Windows Authentication (this is possible since you are logged on to the SQL VM Virtual Machine already. Notice the admin and password field will become greyed out.



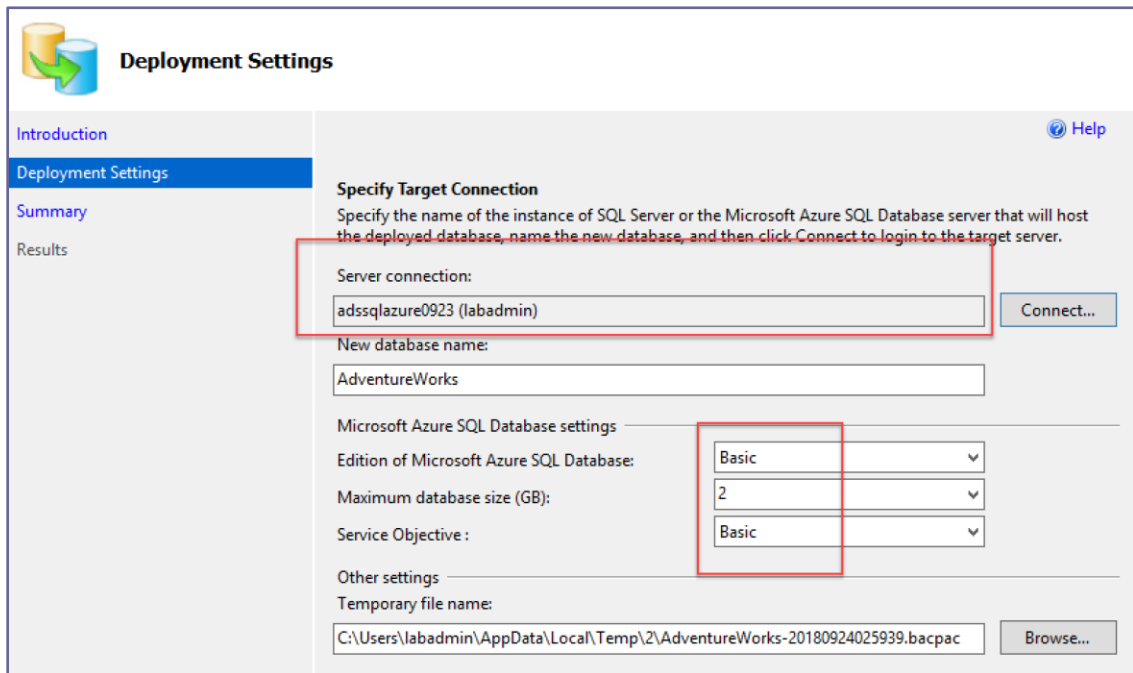
12. Press the **Connect** button.
13. The **Object explorer** shows a successful connection to both databases now. If you open the Databases level, you should see the **AdventureWorks** database.



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 Microsoft 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**



**Deployment Settings**

Introduction  
**Deployment Settings**  
 Summary  
 Results

**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:  
 adssqlazure0923 (labadmin) Connect...

New database name:  
 AdventureWorks

Microsoft Azure SQL Database settings

Edition of Microsoft Azure SQL Database: Basic

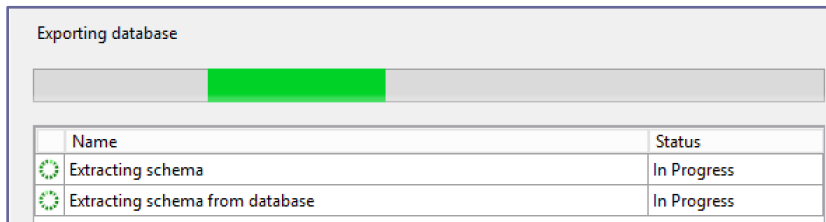
Maximum database size (GB): 2

Service Objective: Basic

Other settings

Temporary file name:  
 C:\Users\labadmin\AppData\Local\Temp\2\AdventureWorks-20180924025939.bacpac Browse...

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

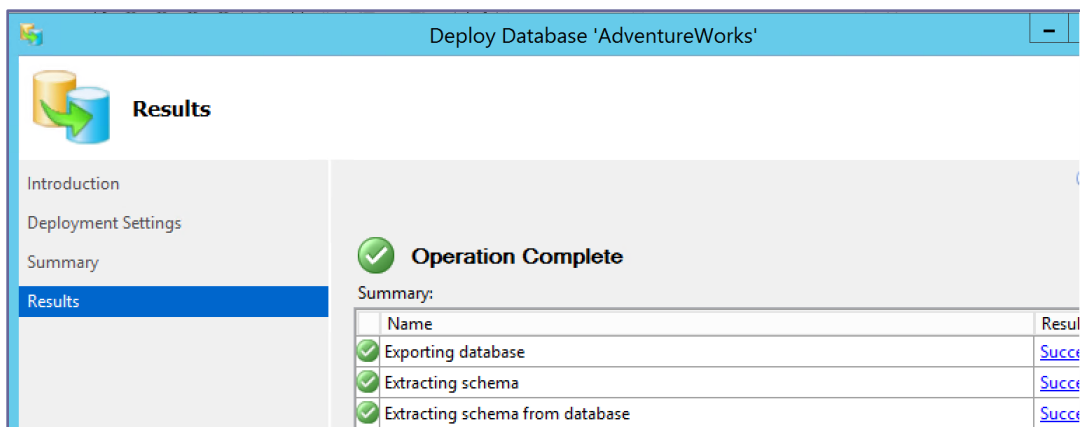


Exporting database

Progress bar: [Green bar indicating progress]

Name	Status
Extracting schema	In Progress
Extracting schema from database	In Progress

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



**Results**

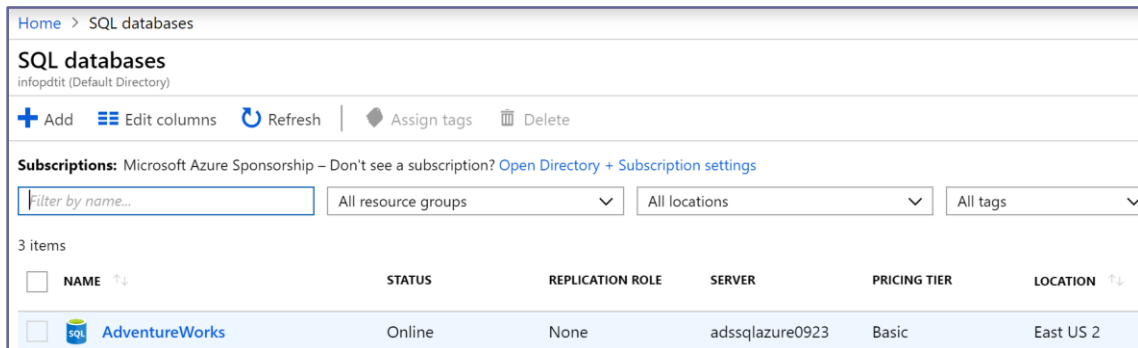
Introduction  
 Deployment Settings  
 Summary  
**Results**

**Operation Complete**

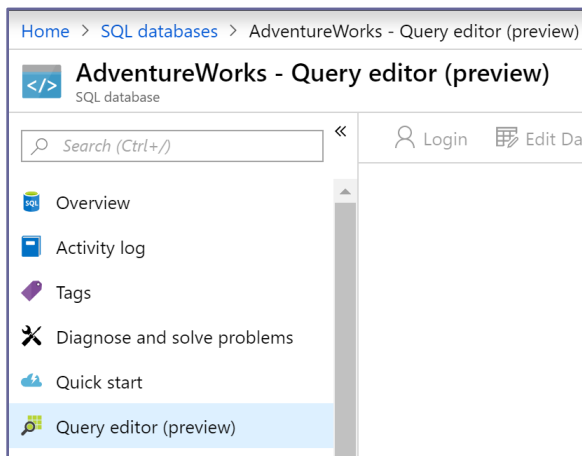
Summary:

Name	Result
Exporting database	Succes
Extracting schema	Succes
Extracting schema from database	Succes

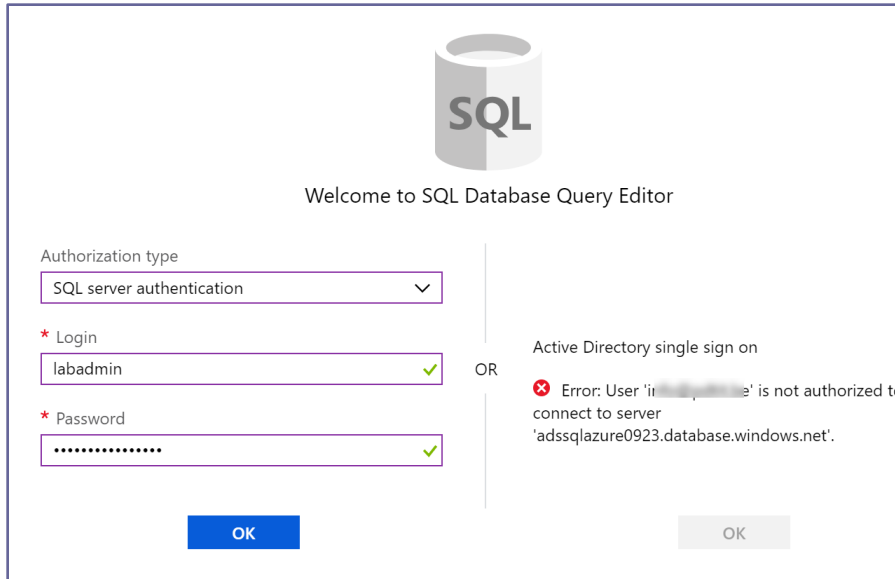
19. Once completed, close the migration window.
20. To validate the database is actually migrated successfully, one can go back to the Azure Portal, and use the Query editor, which is currently in preview, to check on the contents of the database and tables. From the **Azure Portal**, navigate to **all 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.



Query 1 X

Run Cancel query

```
1 select * from production.product
2
```

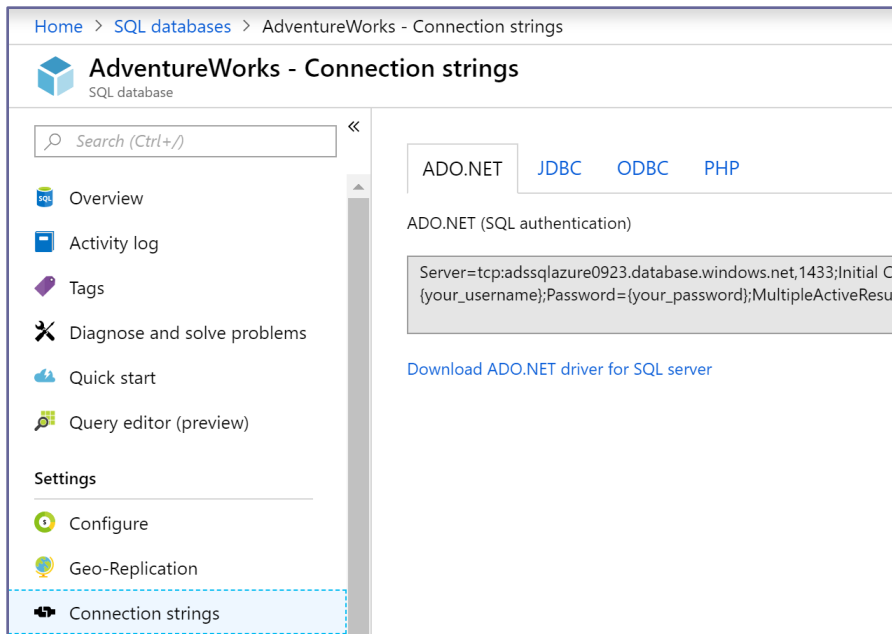
Results Messages

Search to filter items...

PRODUCTID	NAME	PRODUCTNUMBER
1	Adjustable Race	AR-5381
2	Bearing Ball	BA-8327
3	BB Ball Bearing	BE-2349
4	Headset Ball Bearings	BE-2908
316	Blade	BL-2036

### 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.

```

Web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=169433 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=10.0.1.4;initial catalog=AdventureWorks;User ID={your_username};Password={your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="2.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="IsHealthy" value="true" />
  </appSettings>
  <system.web>

```

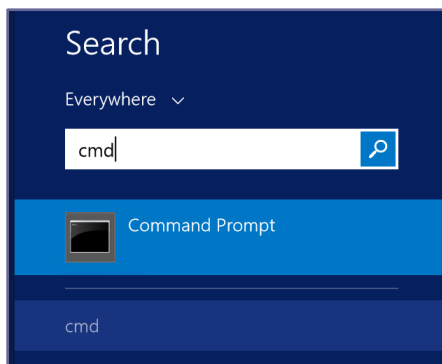
- Go to the section that starts with `<connectionStrings>`. Replace the following settings with the parameters from the Connection String information in the Azure Portal, in both lines starting with `<add name=`:

- `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

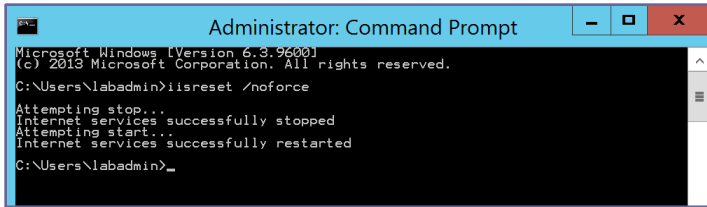
```
<connectionStrings>
<add name="DefaultConnection" connectionString="Data Source=adssqlazure0923.database.windows.net;
initial catalog=AdventureWorks;Uid=labadmin;Password=L@BadminPa55w.rd;MultipleActiveResultSets=True"
providerName="System.Data.SqlClient" />
<add name="AdventureWorksEntities" connectionString="metadata=res://*/Models.AdventureWorks.csdl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

- Save the changes to the web.config file.
- 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.
- From the Start Screen on the WebVM, open a command prompt, by typing "CMD".



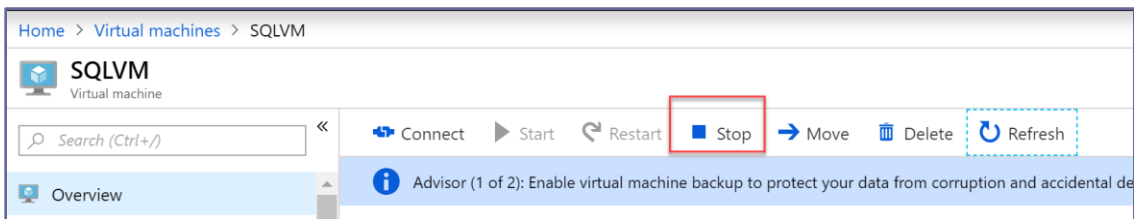
- In the command prompt, run the following command, to restart the IIS Web Server service.

```
iisreset /noforce
```

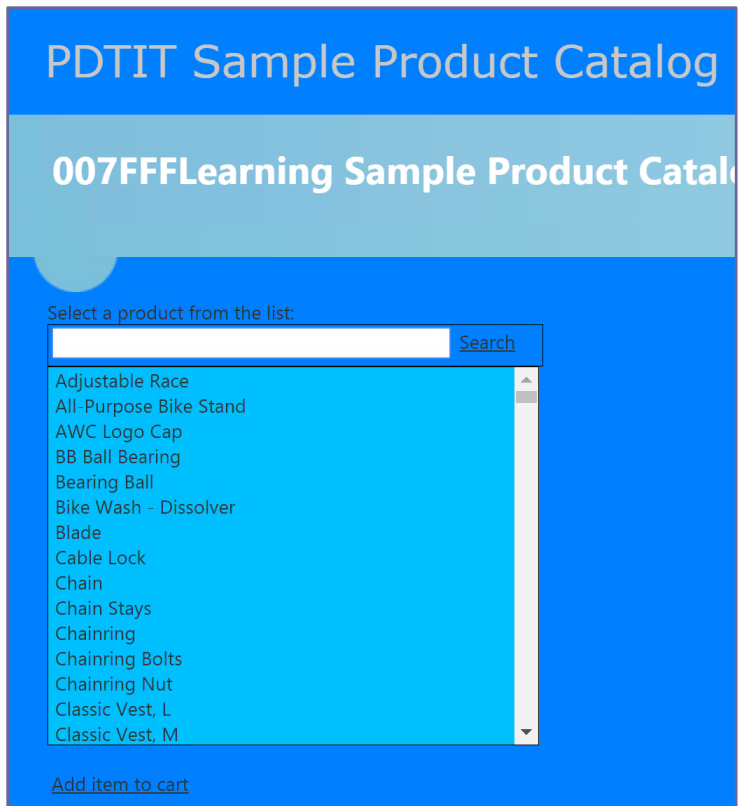


```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\labadmin>iisreset /noforce
Attempting stop...
Internet services successfully stopped
Attempting start...
Internet services successfully restarted
C:\Users\labadmin>
```

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.

## Optional Lab on Azure Database Migration Assistant

Although the lab steps are completed, we want to offer you an optional lab around Azure database migrations, using the Azure Database Migration Assistant.

In short, you will perform the following tasks:

1. Install the Azure Database Migration Assistant on the Web VM
2. Perform an assessment of the to-be-migrated database
3. Run an actual migration process from SQL VM to a new SQL Azure instance
4. Validate the database connectivity using the Azure Portal

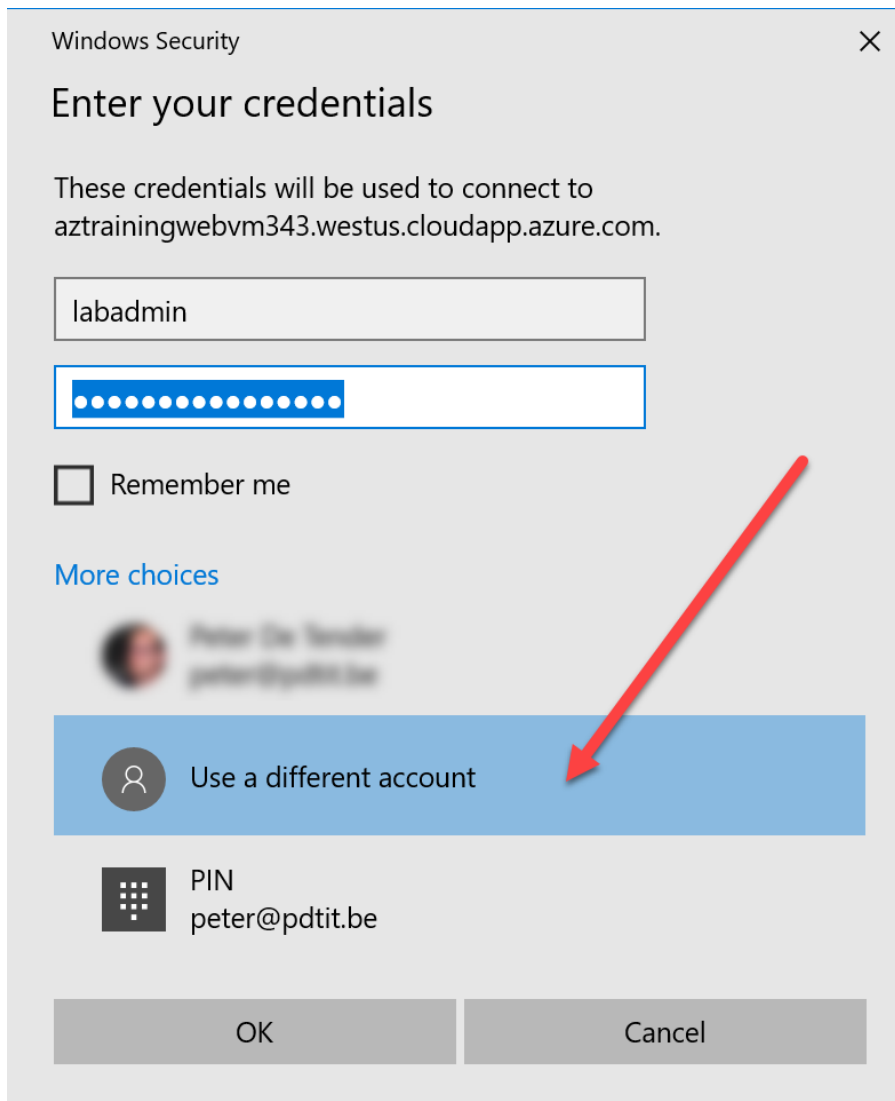
### Task 1: Installing Azure Database Migration Assistant

In this task, you download and install the Azure Database Migration Assistant. We are using the SQLVM in this lab, but you can run this from any Windows server in the same network as the SQL Virtual Machine, meaning, you don't have to install it on the Web Server VM itself.

1. Connect to the WebVM virtual machine using RDP, by **selecting** the WebVM from the **Virtual Machines** section in the Azure Portal followed by **selecting Connect**.



2. In the **Connect** blade, click the **Download RDP File**. Once downloaded, open the file. This will start the **Remote Desktop**, asking for credentials. Here, select **"Use a different account"** and provide the following credentials:  
user account: **labadmin**  
password: [L@BadminPa55w.rd](#)

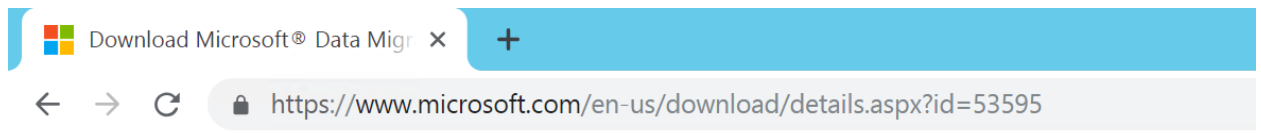


3. When you are prompted for a certificate security warning, select **Don't ask me again...** and press **OK** to continue.



4. Once logged on to the desktop of the WebVM, open the browser, and search for **Azure Database Migration Assistant** download, or connect directly to the following URL:  
<https://www.microsoft.com/en-us/download/details.aspx?id=53595>





## Microsoft® Data Migration Assistant v4.2

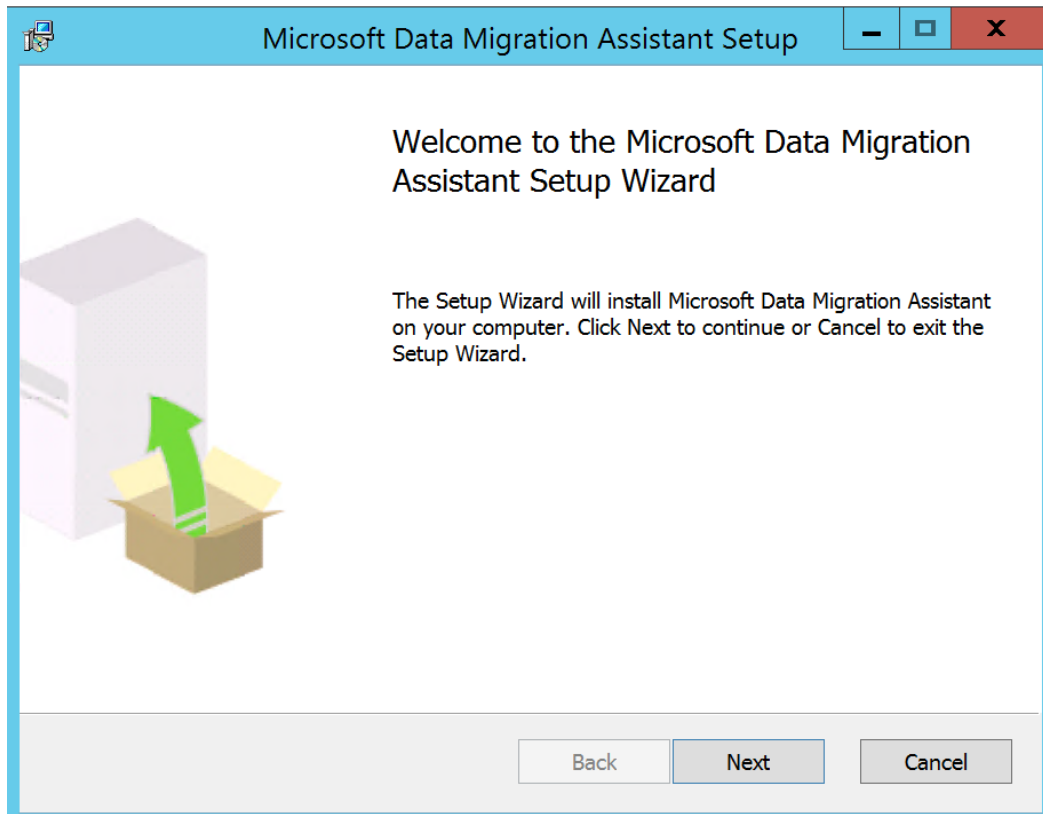
*Important!* Selecting a language below will dynamically change the complete page content to that language.

Language: English

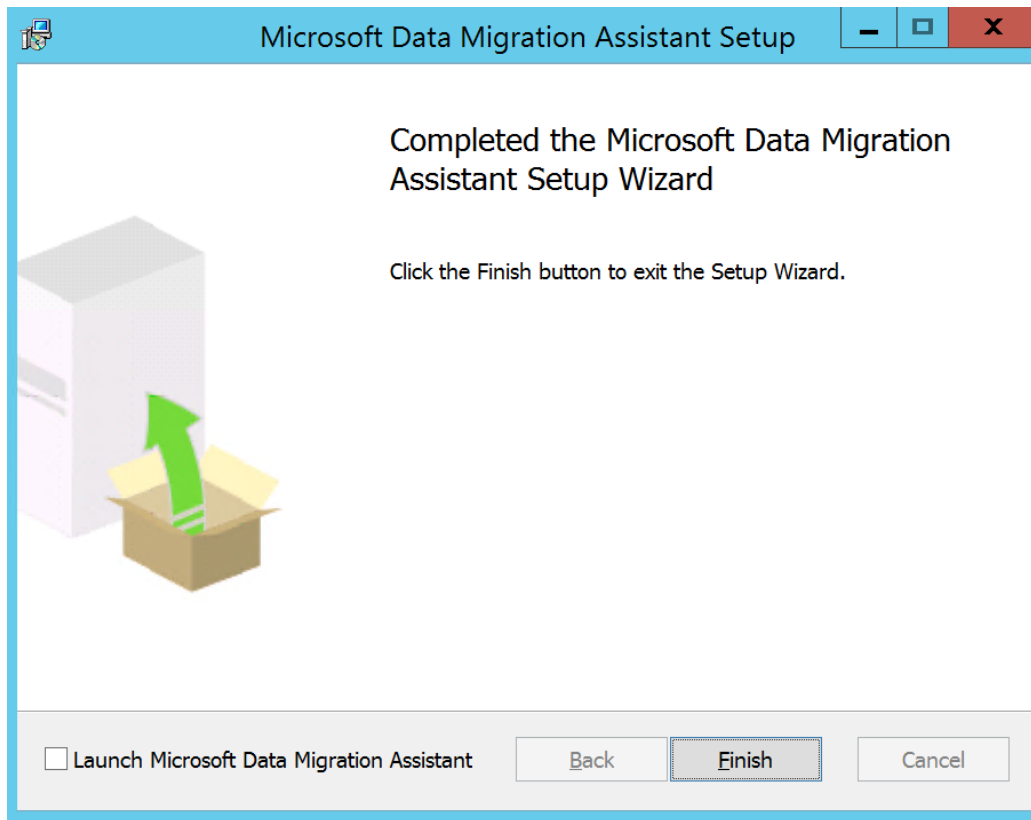
[Download](#)

Data Migration Assistant (DMA) enables you to upgrade to a modern data platform by detecting compatibility issues that can impact database functionality on your new version of SQL Server. It recommends performance and reliability improvements for your target environment. It allows you to not only move your schema and data, but also uncontained objects from your source server to your target server.

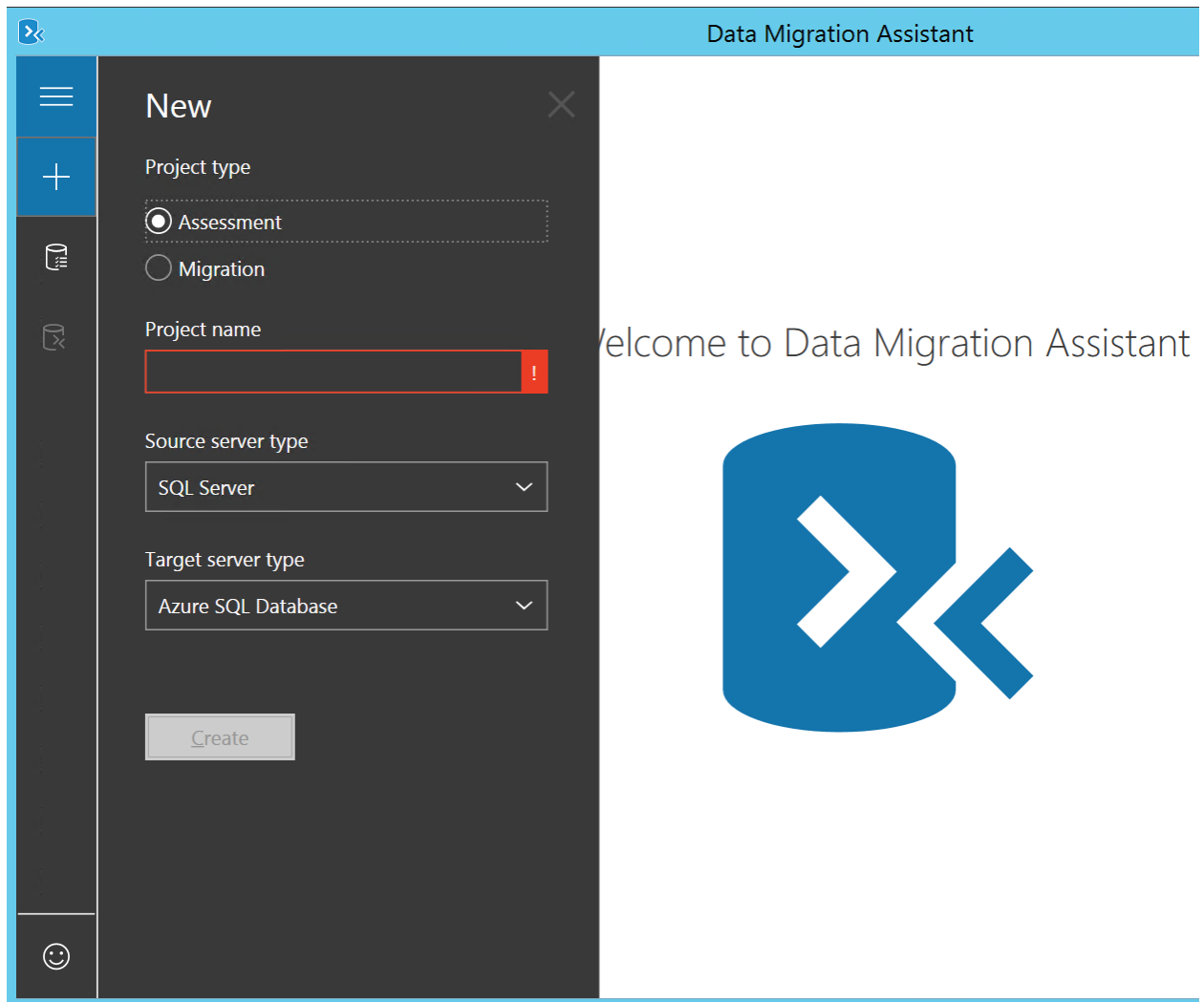
5. Once the download is complete, launch the **DataMigrationAssistant.msi**. Press **Next** to continue.



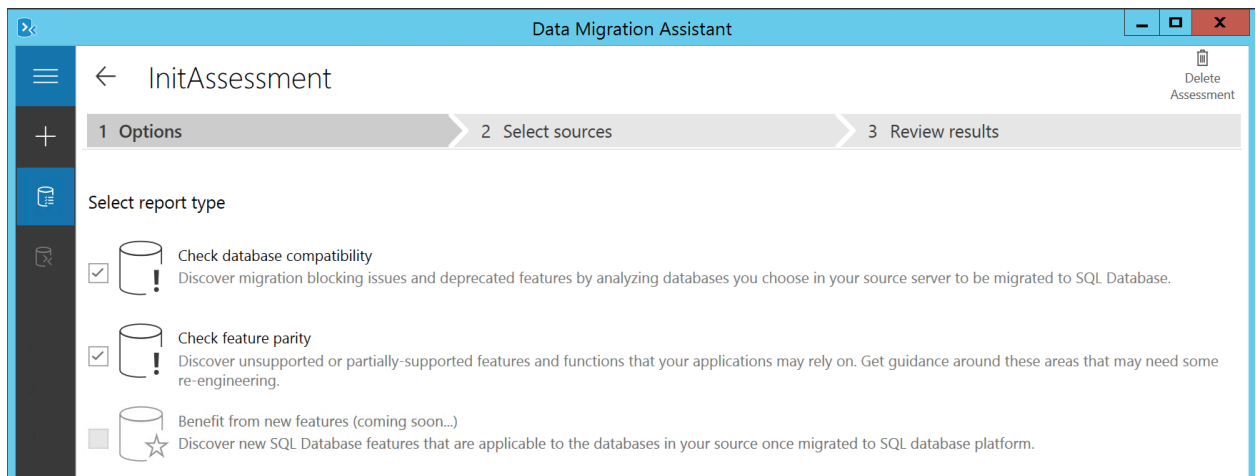
6. Accept the license terms agreement, press Next and confirm by pressing the **Install** button. Wait for the install to complete successfully.



7. If you want to open the DMA tool, select "Launch Microsoft Data Migration Assistant".
8. From **Data Migration Assistant**, select the + on the side to launch a new migration scenario.



9. **We start** by running an **assessment**. Complete the wizard with the following parameters:
  - Project type               **Assessment**
  - Project Name              **InitAssessment**
  - Source Server type       **SQL Server**
  - Target Server type       **Azure SQL Database**
10. This launches the Data Migration Assistant selection window. Here, **press Next** to continue.



11. We now need to connect to our source SQL Server. Therefore, provide the following information in the wizard:

- Server Name **sqlvm**
- Authentication type **SQL Server authentication**
- Username **sa**
- Password **[L@BadminPa55w.rd](#)**

Also flag both options "Encrypt connection" and "Trust certificate".

## Connect to a server

Connect to a server and select sources

Server name

sqlvm

Authentication type

SQL Server Authentication

SQL Authentication credentials

Username

sa

Password

••••••••••••••••

Connection properties

☒ Encrypt connection

☒ Trust server certificate

**SQL Server permissions**

To run the selected advisor(s), credentials used to

Connect

12. Press **Connect** to continue. This brings up the sources list.

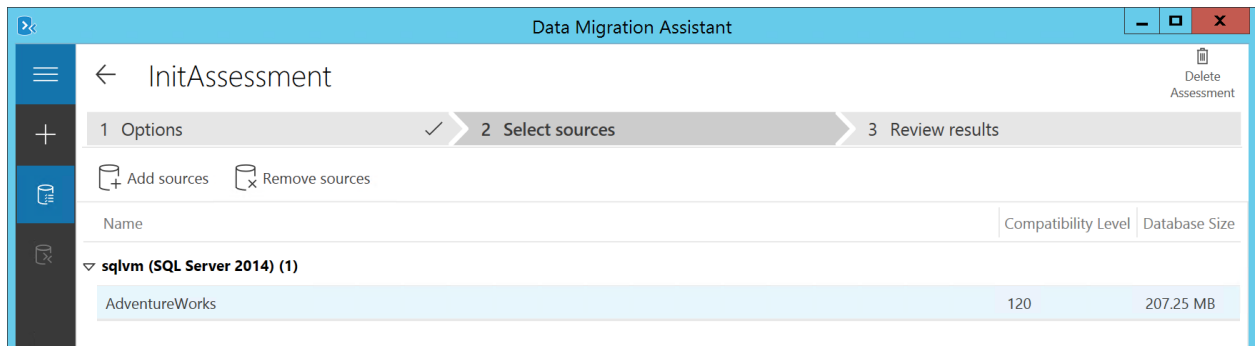
## < Add sources

Connect to a server and select sources

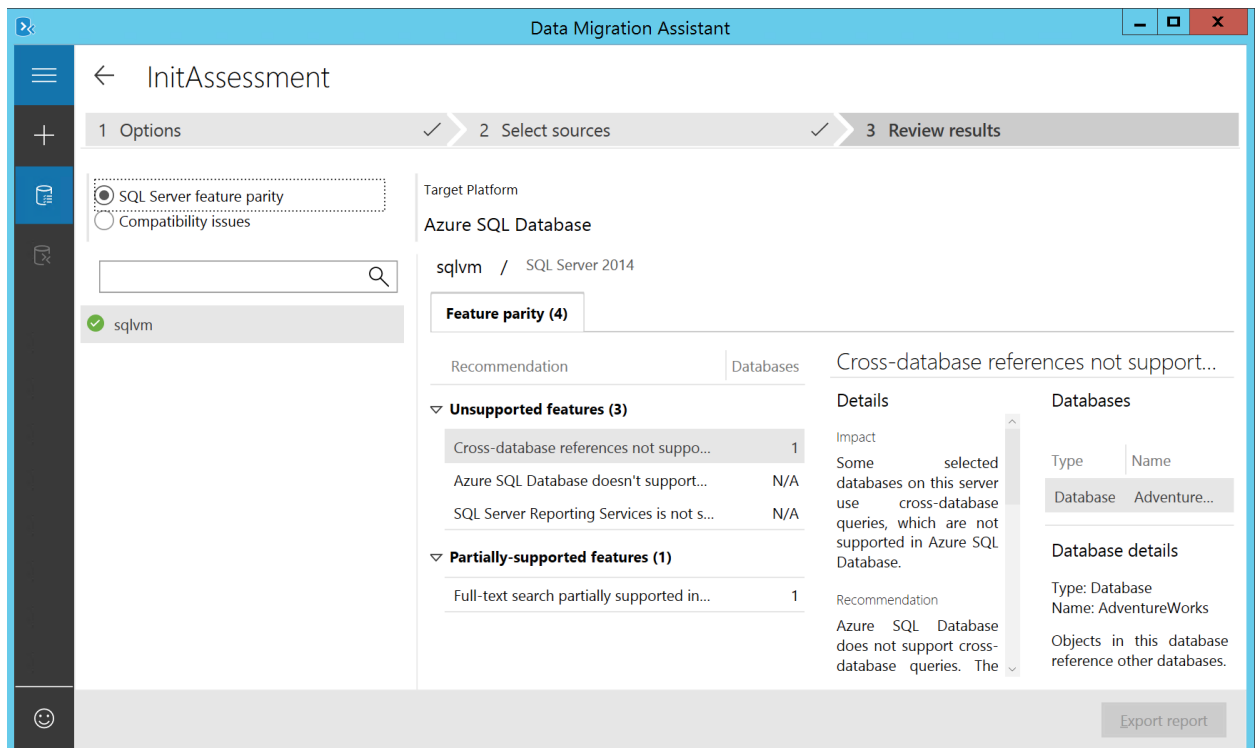
☒ sqlvm

☒ AdventureWorks

13. Select AdventureWorks database, and press **Add** to continue.



14. Next, press the **Start Assessment** button. This runs the assessment, and should take a few minutes to complete. Take note of the several recommendations under **unsupported features** and **partially supported features**.



15. This completes running the assessment. As no **Migration** Blockers came out of the assessment, we can continue with the actual **migration step**.
16. From the left side in the menu, press the **+** button again. This time, select **migration**. Complete the wizard using the following parameters:

- ProjectName      InitMigration

- Source Server Type SQL Server
- Target Server Type Azure SQL Database
- Migration Scope Schema and data

**New**

Project type

☐ Assessment

☒ Migration

Project name

InitMigration

Source server type

SQL Server

Target server type

Azure SQL Database

Migration scope

Schema and data

Create

17. Press **Create** to have the migration job created.

18. From the database connection step, provide the following parameters:

- Server name sqlvm
- Authentication type SQL Server Authentication
- Username sa
- Password [L@BadminPa55w.rd](#)



### Connect to source server

Server name

sqlvm

Authentication type

SQL Server Authentication

SQL Authentication credentials

Username

sa

Password

.....

Connection properties

☒ Encrypt connection

☒ Trust server certificate

#### Source SQL Server permissions

Credentials used to connect to source SQL Server instance must have CONTROL SERVER permission.

Connect

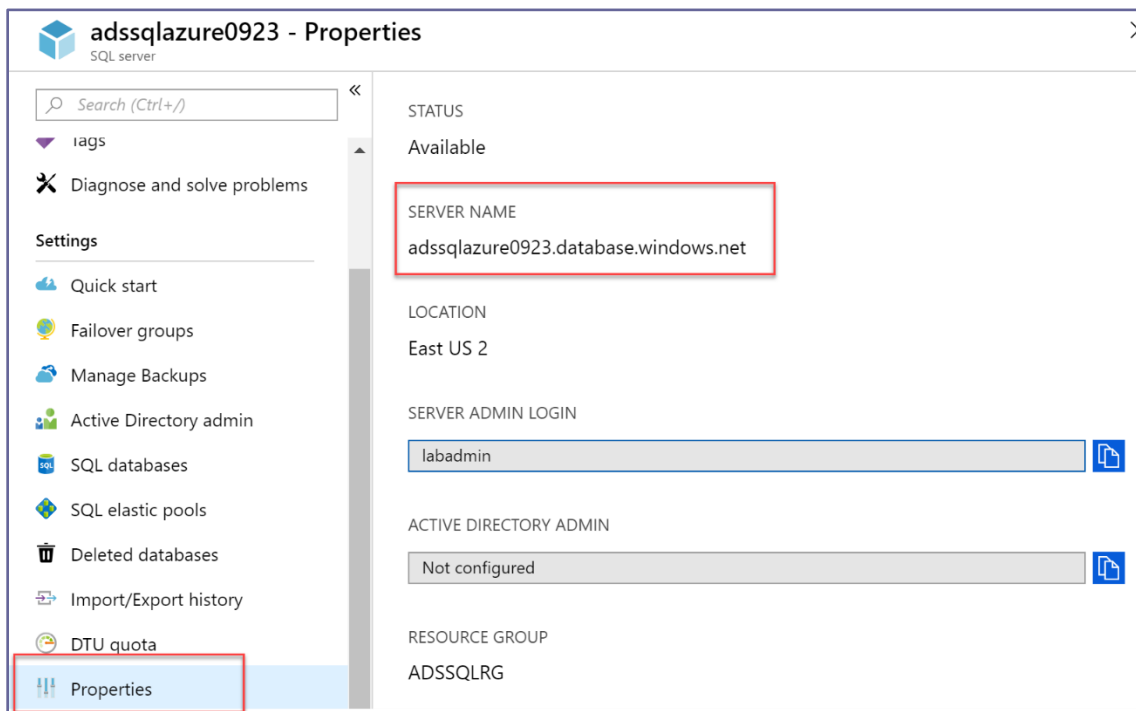
19. Press the **Connect** button, to establish a connection to the SQL VM database.
20. Notice the **AdventureWorks** database getting detected; press **Next** to continue.

Select a single database from your source server to migrate to Azure SQL Database.  
If you skip assessing the databases before migration, DMA will not be able to detect the specific schema objects that m  
Skip this option if you have already done the assessment and addressed the objects with breaking changes prior to the

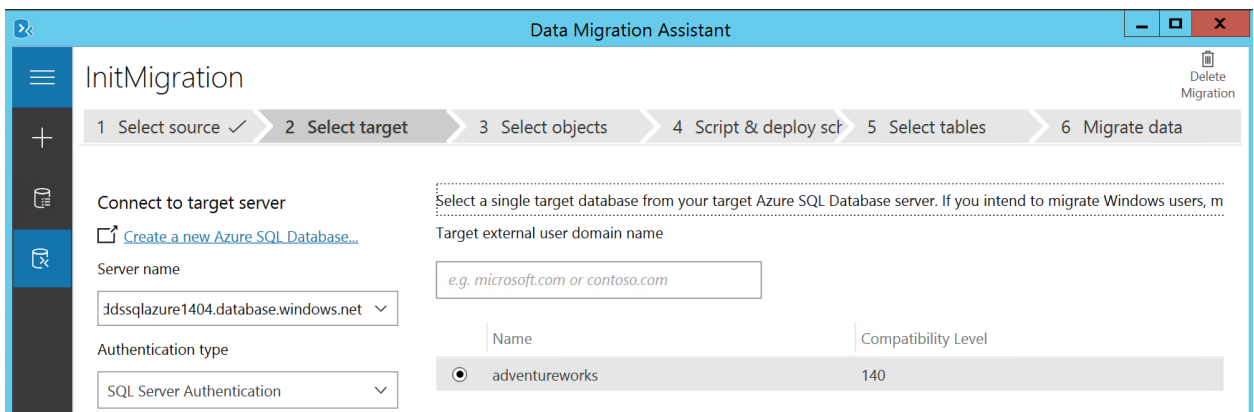
	Name	Compatibility Level	Assess database before migration?
<input checked="" type="radio"/>	AdventureWorks	120	<input checked="" type="checkbox"/>

25. In the **Connect to target server** window, provide the information from the SQL Azure Server you deployed earlier. You can find this in the **Azure Portal**, by browsing to your **SQL Azure**

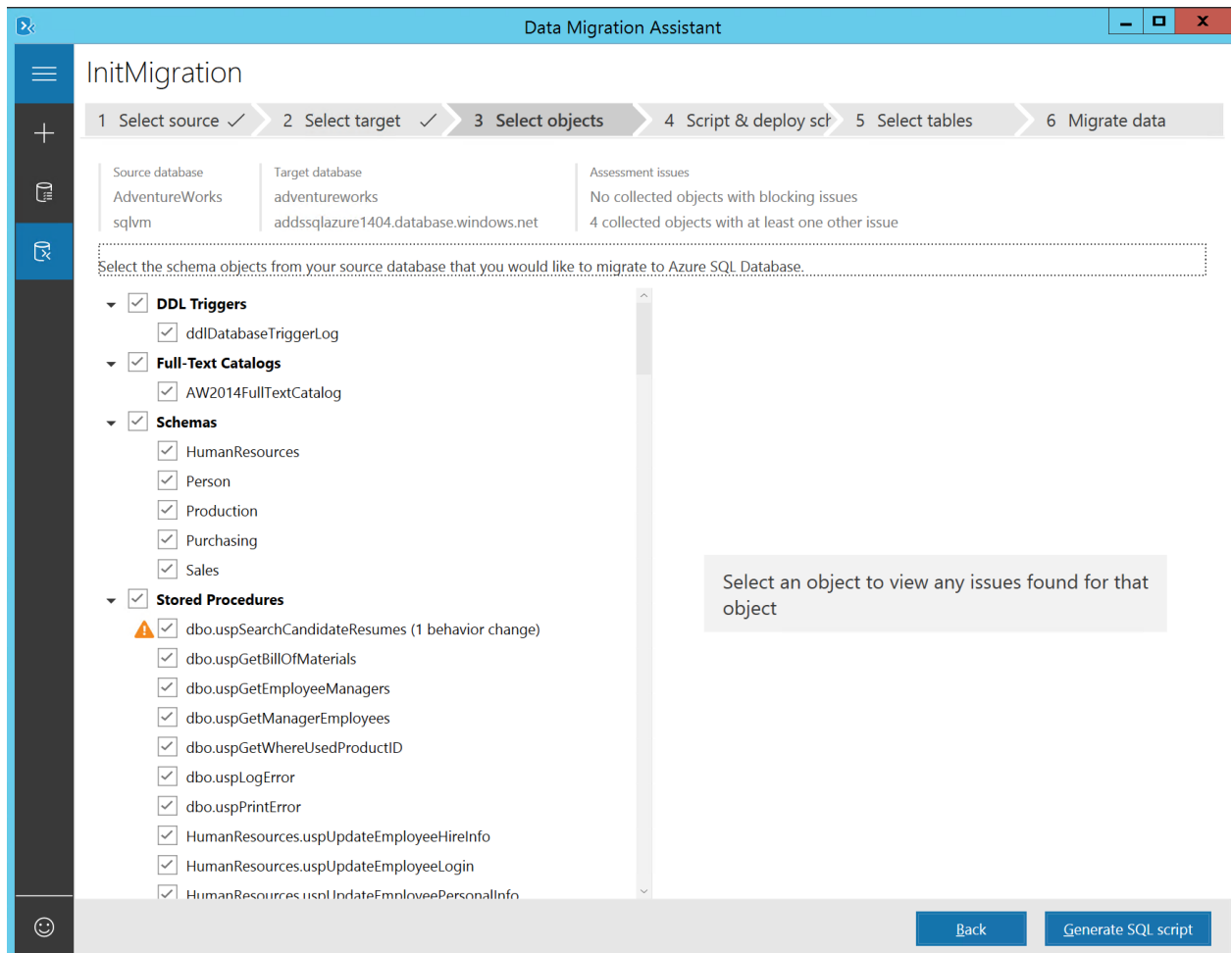
instance, and selecting its properties



21. **Paste** the server name in, and provide the username and password for authentication (account = labadmin; password = L@BadminPa55w.rd)



22. After pressing **Connect**, it will detect the adventureworks database, coming from the previous restore. (Note: if not database gets detected, because you already deleted it, you should create a new one first from the Azure Portal). Press **Next** to continue.
23. This brings you to the **Select Database Objects** step. Validate all the schema objects you want to migrate, and confirm by pressing the **Generate SQL Script** button.



24. This results in a SQL script getting created, and displayed in the body of the migration tool. Perform the actual migration by pressing the **Deploy Schema** button. This kicks off the schema objects creation, running through several tasks during the deployment.

## InitMigration

1 Select source ✓

2 Select target ✓

3 Select objects ✓

4 Script & deploy schema

5 Select tables

6 Migrate data

Source database	Target database	Assessment issues
AdventureWorks	adventureworks	No selected objects with blocking issues
sqlvm	addsqlazure1404.database.windows.net	4 selected objects with at least one other issue

This script was generated for the selected schema objects. Review the script, make edits if necessary, and click "Deploy schema" to deploy to Azure SQL Database. Any selected users were not scripted; these will be migrated separately upon clicking "Deploy schema." SQL logins associated with selected users will be recreated with strong random passwords. You will need to change these passwords and enable them again on the target.

Generated script

Previous issue Next issue Save Copy

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Contains objects'
GO
/***** Object: Schema [Production] Script Date: 4/14/2019 5:31:09 PM *****/
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Production')
EXEC sys.sp_executesql N'CREATE SCHEMA [Production]'
GO
IF NOT EXISTS (SELECT * FROM sys.fn_listextendedproperty(N'MS_Description', N'
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Contains objects'
GO
/***** Object: Schema [Purchasing] Script Date: 4/14/2019 5:31:09 PM *****/
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Purchasing')
EXEC sys.sp_executesql N'CREATE SCHEMA [Purchasing]'
GO
IF NOT EXISTS (SELECT * FROM sys.fn_listextendedproperty(N'MS_Description', N'
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Contains objects'
GO
/***** Object: Schema [Sales] Script Date: 4/14/2019 5:31:09 PM *****/
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = N'Sales')
EXEC sys.sp_executesql N'CREATE SCHEMA [Sales]'
GO
```

Deployment results (97 commands executed..)

Previous error Next error Export

- Executed 91 of 2,236: SET ANSI\_PADDING... Command executed successfully.
- Executed 92 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.
- Executed 93 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.
- Executed 94 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.
- Executed 95 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.
- Executed 96 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.
- Executed 97 of 2,236: IF NOT EXISTS (SELEC... Command executed successfully.

25. Wait for the **schema-migration** to be completed successfully.

```
ys.fn_listextendedproperty(N'MS_Description' , N'
y @name=N'MS_Description', @value=N'Contains objects'
] Script Date: 4/14/2019 5:31:09 PM *****/
ys.schemas WHERE name = N'Sales')
E SCHEMA [Sales]'
```

Command executed successfully.

Executed 2,235 of 2,236: IF NOT EXISTS (SE... Command executed successfully.

Executed 2,236 of 2,236: IF NOT EXISTS (SE... Command executed successfully.

Schema migration completed

Duration: 0h 2m 3s

Back

Redeploy schema

Migrate data

26. Next, set off the data migration, by **pressing** the **Migrate Data** button. This kicks off the actual table migration. Wait for it to complete successfully.

## InitMigration

1 Select source ✓ 2 Select target ✓ 3 Select objects ✓ 4 Script & deploy ✓

71  
Server objects

64  
In-progress

7  
Successful

0  
Warnings

0  
Failed

### Tables (71)

Status	Table name	Migration details
✓	[HumanResources].[Department]	Migration successful. Duration: 0 h...
✓	[HumanResources].[Employee]	Migration successful. Duration: 0 h...
✓	[HumanResources].[EmployeeDepartmentHistory]	Migration successful. Duration: 0 h...
✓	[HumanResources].[EmployeePayHistory]	Migration successful. Duration: 0 h...
✓	[HumanResources].[JobCandidate]	Migration successful. Duration: 0 h...
✓	[HumanResources].[Shift]	Migration successful. Duration: 0 h...
...	[Person].[Address]	<div><div></div></div> 51.0 %
✓	[Person].[AddressType]	Migration successful. Duration: 0 h...
...	[Person].[BusinessEntity]	<div><div></div></div> 96.3 %
...	[Person].[BusinessEntityAddress]	<div><div></div></div> 51.0 %
...	[Person].[BusinessEntityContact]	<div><div></div></div> 0.0 %

27. Wait for the database migration to finish.

InitMigration Delete Migration

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

71  
Server objects

0  
In-progress

71  
Successful

0  
Warnings

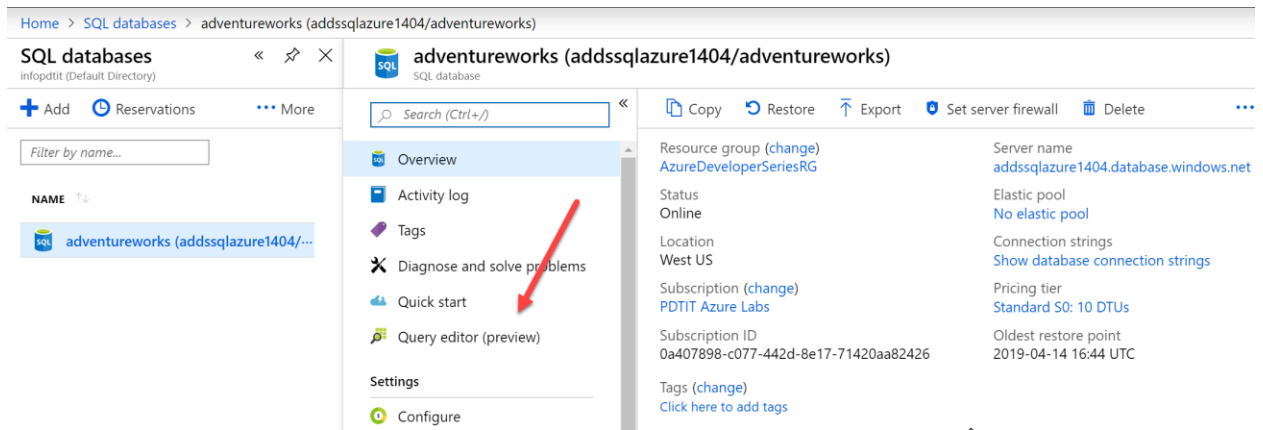
0  
Failed

Source database  
AdventureWorks  
sqlvm

Target database  
adventureworks  
addssqlazure1404.database.wir

Tables (71)

28. Close the Azure Database Migration Assistant and connect to the Azure Portal again. Here, browse to **SQL Databases**, and select the **adventureworks** database we migrated to.



29. **Select Query editor.** Note the **Azure administrative user** has no access rights to the database itself, but it requires SQL authentication.



Welcome to SQL Database Query Editor

Authorization type

SQL server authentication

\* Login

labadmin

\* Password

.....

OK

OR

Active Directory single sign on

✖ Error: User 'info@pdtit.be' is not authorized to connect to server 'addssqlazure1404.database.windows.net'.

OK

30. **Provide the credentials** for the labadmin user (labadmin / [L@BadminPa55w.rd](#)), and connect to the database by pressing **OK**.

adventureworks (labadmin)

Showing limited object explorer here. For full capability please open SSDT.

- Tables
  - dbo.AWBuildVersion
  - dbo.BuildVersion
  - dbo.DatabaseLog
  - dbo.ErrorLog
  - HumanResources.Department**
  - HumanResources.Employee
  - HumanResources.EmployeeDepa...
  - HumanResources.EmployeePayHi...
  - HumanResources.JobCandidate
  - HumanResources.Shift
  - Person.Address
  - Person.AddressType
  - Person.BusinessEntity
  - Person.BusinessEntityAddress

Query 1 X

Run Cancel query

```
1 select * from humanresources.employee
2
```

Results Messages

Search to filter items...

BUSINESSENTITYID	NATIONALIDNUMBER	LOGINID
1	295847284	adventure-works\ken0
2	245797967	adventure-works\terri0
3	509647174	adventure-works\roberto0

31. From the **Tables** view, select the arrow, which opens up all migrated tables. In the **Query** window, enter the following query as an example:

```
select * from humanresources.employee
```

and confirm executing this query by pressing **Run**. This will show you the contents from that specific table.

32. This completes the migration of your SQL VM database using Database Migration Assistant.

## Lab Summary

In this lab, you learned how to deploy and run Azure Database Migration Assistant to perform both an assessment prior to a SQL database migration, as well as for running the actual schema and database content migration task. You also used the Query editor in the Azure Portal to validate the table content of the migrated database.

# Migrating a legacy ASP.NET 2-tier application to Azure using Container Services

Hands-On-Labs step-by-step guides

Prepared by:

**Peter De Tender**

CEO and Lead Technical Trainer  
PDTIT and 007FFFLearning.com

@pdtit

@007FFFLearning

Version: April 2019





